Heterogeneous network architectures

a layered approach for modeling key building blocks for continuing network evolution

Christiansen, Henrik Lehrmann

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

Heterogeneous network architectures

- a layered approach for modeling key building blocks for continuing network evolution

Ph.D. thesis

Submitted to Research Center COM at the Technical University of Denmark
in partial fulfillment of the requirements for the Ph.D. degree.

Henrik Christiansen

August 2004

Research Center COM

Technical University of Denmark

Kgs. Lyngby

Denmark

Abstract

Future networks will be heterogeneous! Due to the sheer size of networks (e.g., the Internet) upgrades cannot be instantaneous and thus heterogeneity appears. This means that instead of trying to find *the* solution, networks should be designed as being heterogeneous. One of the key requirements here is flexibility.

This thesis investigates such heterogeneous network architectures and how to make them flexible.

A survey of algorithms for network design is presented, and it is described how using heuristics can increase the speed. A hierarchical, MPLS based network architecture is described and it is discussed that it is advantageous to heterogeneous networks and illustrated by a number of examples.

Modeling and simulation is a well-known way of doing performance evaluation. An approach to event-driven simulation of communication networks is presented and mixed complexity modeling, which can simplify models, is introduced.

Modeling and simulation is then used to evaluate the behavior of adaptation devices in the context of heterogeneous networks - devices that can interconnect network domains employing diverse technologies. The simulation shows how queuing disciplines impact delay profiles. TCP and application behavior on top of such adaptation devices are also investigated by simulation.

Finally, a new concept for packet forwarding is introduced and modeling of this scheme is presented. The simulation results show that the scheme is feasible to use in future networks.

All in all these issues investigated is part of what is needed for getting the required flexibility into future, heterogeneous networks.

Resumé (pa dansk)

Fremtidens net vil være heterogene! Grundet selve størrelsen af net (f.eks. Internettet) kan opgraderinger ikke ske øjeblikkeligt og derfor opstår denne heterogenitet. Derfor bør man i stedet for at lede efter én løsning, hellere designe fremtiden net så der tages højde for denne heterogenitet. Et af hovedkravene her er fleksibilitet.

Denne afhandling undersøger sådanne heterogene net arkitekturer og ser på hvordan de kan gøres fleksible.

En oversigt over algoritmer til net design vil blive præsenteret og det bliver beskrevet hvordan man ved hjælp af heuristikker kan øge hastigheden af beregningerne. En hierarkisk, MPLS baseret net arkitektur bliver derefter beskrevet og dens fordele i forbindelse med heterogene net arkitekturer bliver beskrevet sammen med en række eksempler.

Modellering og simulering er en velkendt metode til performance evaluering. Her præsenteres en metode til *event-driven* simulering af kommunikationsnet og der fremlægges en metode til at simplificere modellerne og dermed opnå hurtigere simuleringer.

Herefter anvendes modellering og simuleringsmetoden til at undersøge hvordan *adaptation* enheder opfører sig i netværkssammenhæng – disse enheder anvendes til at forbinde netværksdomæner med forskellige teknologier. Simuleringerne viser hvordan schedulering har indflydelse på forsinkelse i nettet. Det undersøges endvidere hvilken indflydelse det har på TCP og applikationerne.

Desuden fremlægges et nyt koncept for hvordan man kan sende pakke gennem nettet. Denne metode modelleres også og simuleringerne viser, at metoden vil være brugbar i fremtidens net.

Summa summarum er alle de emner der diskuteres en del af et samlet hele: det er brikker i den fleksibilitet, der kræves af fremtidens net.

Acknowledgments

Even though only one name appears in the author list of the thesis a number of people have contributed considerably to its preparation. The list of people is very long, so I have selected a few to include here. If I have left anyone out, it is not a deliberate attempt of being rude - it is simply a consequence of bad memory and trying to save paper…

Thanks to Lars Dittmann my supervisor, for allowing me to work in a one-of-a-kind research group. His unique mentoring style has taught me what flexibility really means and that things don't have to be done in the usual way.

Thanks to the networks group, which I have benefited immeasurable from being part of – to Michael Berger, with whom I had numerous discussions, from which the hierarchical DAVID architecture is a result, to Henrik Wessing for the work we did on the key routing scheme and to all the rest of you for the work, the good spirit and all the fun. Lunchtime discussions are never forgettable moments.

Last but not least thanks to my girlfriend Christine Christiansen for bringing more music into my life, for ensuring that my life was never too network-centric and for playing so beautifully for me to cheer me up when I was grumpy during the thesis work.

Henrik Christiansen, August, 2004

Publications

Conference papers:

[p1] M.Pióro, T. Stidsen, A Glenstrup, C. Fenger, **H.Christiansen,** *"Design problems in robust optical networks"*, Networks 2000, Toronto Canada, September 2000

[p2] **H.Christiansen**, *"A novel switching concept for MPLS"*, poster, Opticom 2000, Dallas, USA, October 2000

[p3] H. Wessing, T. Fjelde, **H.Christiansen**, L. Dittmann, *"Novel scheme for efficient and cost-effective forwarding of packets in optical networks without header modification"*, OFC 2001, Anaheim, USA, March 2001.

[p4] **H.Christiansen**, *"Using OPNET To Compare and Analyze Different Traffic-Bundling Schemes",* OPNETWork 2001, Washington DC, USA, August 2001

[p5] L.Dittmann, **H.Christiansen**, M. Berger, *"Hierarchical MPLS – a scalable approach for efficient resource administration in multi-technology networks"*, NOC 2001, Ipswich, England , June, 2001.

[p6] **H.Christiansen**, *"Modeling GMPLS domains in MPLS networks"*, OPNETWork 2002 , Washington DC, USA, August 2002.

[p7] **H. Christiansen**, M. Berger, *"Novel, hierarchical, MPLS-based network architectures and their role in migration strategies towards future, optical, packet switched networks"*, CIIT 2002, St. Thomas, USVI, November 2002.

[p8] **H. Christiansen**, H. Wessing, *"Modeling GMPLS and optical MPLS networks"*, ICT'2003, Papeete, Tahiti, February 2003.

[p9] M. Berger, **H. Christiansen**, B. Mortensen, R. Jociles-Ferrier, *"Hierarchical Electro-Optical Packet Network Architecture"*, IST2003, Isfaran, Iran, August, 2003.

[p10] E. Dobranowska, **H. Christiansen**, *"Automated topology modeling with OPNET"*, OPNETWork 2003, Washington DC, USA, August 2003.

[p11] L. Staalhagen, **H. Christiansen**, *"TCP traffic in Satellite systems"*, OPNETWork 2003, Washington DC, USA, August 2003.

[p12] **H. Christiansen**, L. Staalhagen, *"Resource administration in satellite systems"*, OPNETWork 2003, Washington DC, USA, August 2003.

[p13] L. Staalhagen, **H. Christiansen** *"Teaching Network Modeling and Simulation using OPNET Modeler"* , OPNETWork 2004, Washington DC, USA August, 2004

[p14] **H. Christiansen**, S. Van Cauwenberge *"A tool for GPRS end-to-end performance modeling"*, OPNETWork 2004, Washington DC, USA August, 2004

Journal papers:

[p15] **H. Christiansen**, T. Fjelde, H. Wessing, *"Novel label processing schemes for MPLS"*, Optical Networks Magazine, November/ December 2002.

[p16] H. Wessing, **H. Christiansen**, T. Fjelde and L. Dittmann, *"Novel scheme for packet forwarding without header modification in optical networks"*, IEEE Journal of Lightwave Technology, vol. 20 #8, August 2002, pp. 1277-1283.

[p17] O.M. Lauridsen, E. Agertoft, **H. Christiansen**, *"EDGE upgrades must consider quality of service"*, WirelessEurope, Issue 30, January, 2004

Public reports / deliverables for EU projects:

[p18] O. Marmur, T. Muzicant, **H.Christiansen**, *"Network Architecture and System Requirements"*, METEOR deliverable D05, 2001,

[p19] L. Dittmann, **H. Christiansen**, P. Vogel, A. Kapovits, *"Project objectives and benchmark measures for next generation core and metro networks"*, NGNI NGPN deliverable D1, October 2002.

[p20] **H. Christiansen**, *"Topologies and architectures for next generation core and metro networks"*, NGNI NGPN deliverable D4, October 2002.

[p21] N. Le Sauze, D. Chiaroni, M. Nord, M. Berger, **H. Christiansen**, J. Fernandez-Palacios, J. Lobo, D. Careglio, J. Solé-Pareta, S. Spadaro, A. Rafel, A. Hill, S. Sygletos, H. Skoufis, A. Stavdas, H. Lønsethagen, T. Olsen, F. Callegati, F. Neri, A. Bianco, G. Galante,

M. Mellia, *"Network concepts validation and benchmarking"*, DAVID deliverable, December 2003

[p22] A. Ackaert, S. Demaesschalck, D. Colle, P. Demeester, M. O Mahony, C. Politi, M. Falch, D. Saugstrup, K.E. Skouby, R. Tadayoni, **H. Christiansen**, J. Soler, L. Dittmann, D. Erasme, G.Rodriguez, C. Minot, I. Fsaises, J. Faber, G. Grosskopf, E. Patzak, H. Thiele, M. Schlosser, J. Vathke, S. Rao, P. Vogel, L. Tuomi, J.C. Point, *"First, combined, report on the multi- technological and multi-disciplinary analysis of the 'broadband for all' concept."*, BREAD deliverable, March 2004.

Contents

1. Introduction

"There will come a time when you believe everything is finished. That will be the beginning."

L. L'Amour

Ubiquitous data services with the ability to provide access to information, voice and multimedia sessions are a much-vaunted feature in future networks. One way to realize this is to use IP as the glue bridging the gap between diverse access technologies and provide a consistent interface to applications. This thesis gives a survey of possible solutions to the problems researchers face in their quest for pervasive services – mainly focusing on the evolving Internet.

The Internet relies on a few very basic principles: the self-describing datagram-packet, diversity in technology and global addressing [Clark2002]. Although there have been attempts to change these fundamental properties [Stoica2004][Smi2004] this thesis will assume that these properties prevail and look at how networks can evolve within these limits.

A network is made from a number of components. Communication media - electrical wires, optical fibers or the air in the case of wireless communication – are commonly referred to as network links. These links interconnect the network nodes, which are switching devices capable of analyzing and handling information, or traffic, at branching points. Together, network nodes and links and the way in which they are interconnected form the network topology. Information exchange on links, also known as transmission, is limited by the available transmission technologies. Transport of traffic across several links and nodes is limited also by the capabilities of the nodes – network technologies. On top of all this there are protocols that govern how traffic flows within the network - the combination of topology and protocols is the *network architecture*. The capacity of a network is a measure of the information transport resources of a network. The capability of a network determines the services that the network is able to offer and clearly depends on the capacity. Network capacity and capability depend on the network architecture.

The need for network capacity is growing, especially due to the increasing number of users of the worldwide Internet. Consequently, a vast number of researchers and companies are working on high-speed technologies for the

network links and nodes. These new methods are almost exclusively based on optical technologies. A number of mature technologies for high capacity transmission e.g., wavelength division multiplexing (WDM) combined with high bit rates on optical fibers, are now available and are being implemented; hence the sheer transmission capacity of networks is fulfilling the requirements. In addition, optical technologies are also brought to the network nodes in order to increase their capacity and thus eliminate them as bottlenecks, which they clearly are today. However, contrary to the links, which are merely information pipes, network nodes are devices performing complex operations that are clearly beyond the capabilities of optical technologies. Thus, increasing the networks' capacity is insufficient in order to create future networks.

1.1. Requirements to future networks

Before getting into the technical details it might be beneficial to look at some overall requirements to the networks of the future. It is hard to set up realistic, general requirements to future networks, because who knows how they will evolve? This thesis has no intent to act like a crystal ball, predicting the future – it's unpredictable! However, exactly this unpredictability gives rise to one requirement: *flexibility*. New networks must be flexible so that they can migrate to a state, which fulfils whatever requirements future needs might impose on them. This thesis tries to identify properties that will be common to *all* future networks!

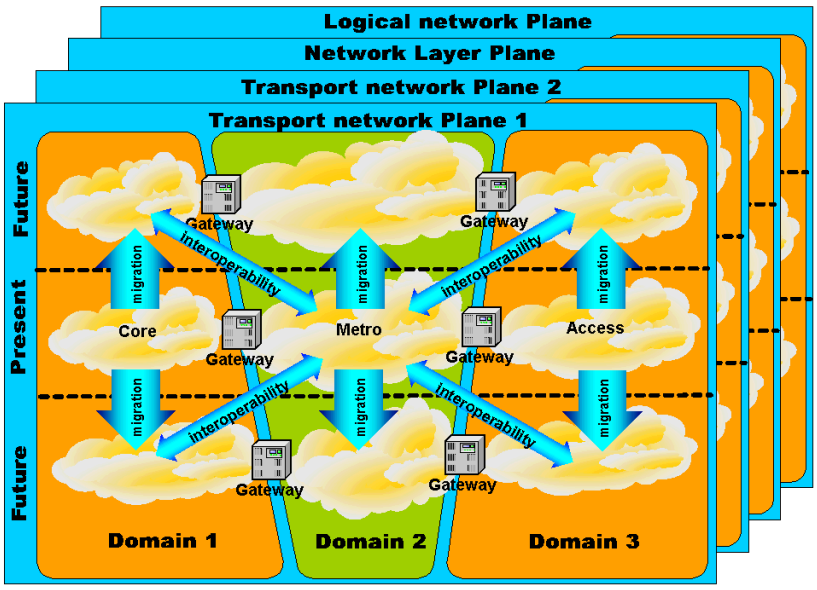


Figure 1: Network evolution (user plane)

The figure above (Figure 1) depicts the general evolution of networks. Presently, we have the network split into three domains: core, metro and access. This separation or subdivision of networks might not pertain – as illustrated (moving either upwards or downwards in the figure), the size of the domains might shrink or grow and the functionality as well as the names might change completely. Hence the generic terms domain 1, domain 2 and domain 3 in the figure. The migration in those three domains might not happen at the same pace, but interoperability with legacy domains must be assured. Additionally, the whole story, subdivision into domains, functionality, migration path and speed is not necessarily identical for the various planes in the network. Depending on application, e.g., the control plane might evolve differently from the management plane. Again, however, interoperability with legacy networks and flexibility (which can be seen as interoperability with *future* networks) must be assured.

The figure shows a number of planes. The two *transport network* planes illustrate two different transport technologies (e.g., protocols or physical bit transport methodologies). The figure should not be understood in the way that exactly two different transport technologies are always required, rather in the way that different technologies (could also be more than two) can coexist within the network. In addition there are a network plane and a logical network plane.

The figure could be repeated for the control and management parts of the network as well, illustrating that the technologies and structures not necessarily have to be the same for the user, control and management plane, respectively.

Requirements to future networks could be summarized as:

- Transparency (signal format, protocols independence)
- Traffic engineering capabilities (including support for mobility)
- End-to-end control / QoS support
- Flexibility

New frontiers (such as: ad hoc networks, peer to peer networks, storage area networks (SAN), pervasive computing, sensor networks, wearable computing etc.) makes these requirements even more pronounced

1.1.1. Convergence and divergence – simultaneously!

From a user or application point of view the networks evolves towards more homogenous networks, such as that based on the IP protocol. Thus from an application point of view the networks converge. However, when dwelling

into the networks' architectures and the various protocols and transmission technologies hidden here the picture is somewhat different. New proposals pop up constantly and opposed to the old days where equipment vendors were forced to obtain consensus on some standards in order for their devices to interoperate. Today the requirements for common standards have been somewhat loosened up because of the IP convergence. Thus networks are undergoing a convergence and divergence at the same time.

1.1.2. Heterogeneous networks

This diversity creates heterogeneous networks. The networks are heterogeneous horizontally in terms of various protocols that must be able to interact. Additionally, vertical heterogeneity is seen when different technologies must interoperate.

1.2. Thesis outline

The composition of this thesis is as follows: Chapter 2 *"Technology foundation"* deals with some background information about technologies. A number of issues are identified and this sets the scene for the rest of the thesis. Chapter 3 *"Network resource administration"* treats some aspects of QoS and mainly focuses on optimization techniques for network capacity planning. Chapter 4 *"Network architectures"* introduces a hierarchical network architecture and discusses how future networks can benefit from it. Chapter 5 *"Performance evaluation by simulation"* is about simulation and about methods for doing them fast and right. Particularly, mixed complexity modeling is introduced as a way of simplifying models and thus speed up simulations. This technique will be used in chapter 6 *"Modeling node behavior"*, which covers how to model the network impact of specific node behavior. Finally, chapter 7 contains summary and conclusions.

Throughout the thesis, references to publications that I have co-authored are denoted by [pXX] and refer to the publications list in the beginning of this thesis. As can be seen just from reading the paper titles a rather broad area of topics have been covered (from network planning via network architectures and wireless networks to detailed modeling of network node devices) and will also be covered in this thesis. Hence, not every single detail is included here, but references to further descriptions in the literature are given. For the convenience of the reader at the end there is a list of frequently used abbreviations and acronyms.

1.2.1. A note on novelty

The work presented in this thesis is primarily based on work carried out within EU research projects. This short section highlights the novel work presented in this thesis and when the results is based on joint work, describes my role and contribution to the work done.

The work on optimization and heuristics was done with Christians Fenger, Arne Glenstrup and Thomas Stidsen. [p01]

The work on hierarchical network architectures was done together with Michael Berger as part of the work we did within the DAVID project.[p05][p07]

The work on key routing was made in collaboration with Henrik Wessing based on my original idea and developed further during Wessing's master Thesis (which I supervised).[p02][p03][p08][p15][p16]. The initial work on basic traffic aggregation was done with Michael Berger while doing the work on the DAVID network architecture [p09]. The work on modeling of the Inmarsat satellite communication systems was done together with Lars Staalhagen. [p11][p12]

I also spent some time on conceptual modeling issues. The modeling and simulation is entirely my work. [p04][p06]

Everything is layered! This thesis is about how to exploit the layered nature of networks when doing resource reservation, when building flexible network architectures and when modeling networks. Layering is a flexible approach to flexible networks – the kind of networks needed for the future!

2. Technology foundation

"The bend in the road is not the end of the road unless you refuse to take the turn."

Anon

Networks are complex structures built from a high number of devices using a number of technologies. Obviously the technologies selected impacts the properties of the entire network and hence given a set of network requirements the selection of suitable technologies is not straightforward.

This Chapter aims at giving a snapshot overview of a number of technologies that are relevant for building flexible network architectures and thereby identifying the issues that are investigated in further detail throughout this thesis.

The terminology used throughout this thesis uses technologies to denote transmission technologies and protocols to refer to network layer protocols and above. Thus technologies indicate OSI layer 2 and below, protocols OSI layer 3 and above.

2.1. Protocols / horizontal heterogeneity

To illustrate the diversity in networking protocols this section highlights a number of salient features characterizing the protocols in today's networks.

2.1.1. Protocol stacks

A crucial part of any network architecture is the protocols stack. Protocol functionality has been subdivided into layers in a protocol stack in order to separate functionality and to enable independent research/development in each layer. Each plane in a network uses its own protocol stack. This section focuses on the user plane.

In many core networks, an IP/ATM/SDH/WDM network architecture is used still today, as opposed to a de-layered solution with a collapsed protocol stack of IP/adaptation layer/WDM. Table 1 below contains brief descriptions

of IP, ATM, SDH and WDM, respectively. For a more in-depth coverage, please consult the references.

Technology / protocol	Characteristics
IP	Connection-less, packet switched technology using variable length packets. Widespread use (e.g., in the Internet) with support on almost any client platform. No resource reservation in the network and thus only best-effort transport. Efficient use requires compliant terminals with a suitable transport protocol (such as TCP [rfc793]) that implements a back off strategy to resolve network contention. [rfc791] [rfc793] [rfc3168]
ATM	Connection-oriented, packet switched technology with fixed length packets (cells). Widespread use, but usually hidden to the user. Uses explicit resource reservation to provide QoS guarantees on each connection. Terminals are monitored (policing) at the edge of the network to prevent that excessive network load causes network congestion. [I.361]
SDH	Connection-oriented, circuit switched, multiplexing technology based on TDM. Subdivides in a hierarchical manner a transmission line's total capacity into smaller chunks of rather fine granularity. Build-in protection mechanisms and supervision for interaction with the management system. [G.803]
WDM	Transmission of several wavelengths simultaneously on the same fiber. A way to exploit the enormous inherent transmission capacity of the optical fiber. Enables the use of ADMs. Provides a coarse-grained, raw bit transport mechanism. Depending on the density of the wavelengths this is called either CWDM or DWDM [D3]

Table 1: Overview of the salient features of IP, ATM, SDH and WDM, respectively

However, the advantages of using the full IP/ATM/SDH/WDM solution are that the attractive features from each layer can be combined. WDM brings the ability to exploit as much as possible of the fiber bandwidth, SDH brings protection/restoration capabilities, ATM comes with support for QoS guarantees while IP provides interaction with the majority of clients. All in all a full featured set of properties that are required for a network. One problem

though, is that IP is unaware of the QoS capabilities of ATM and thus still only provides best-effort services. ATM is used more as a tool for network providers to manage flows and bandwidth on a larger scale. In that respect it is a huge benefit compared to pure WDM, because it offers a much finer granularity in bandwidth management.

It has been argued [Foi2001] that the SDH/ATM/IP stack is way too complex because it contains redundant functionality. Hence, there has been done (and still is) a lot of work on collapsing this protocol stack into what is usually referred to as *IP over WDM* or *IP over optics* (although this is a rather silly expression because neither WDM nor optics are protocols. However, there seems to be a trend towards including the physical layer in the protocol stack. This way of describing layered structures has for instance created the term: *radio over fiber*). [Gha2000] [Foi2001]

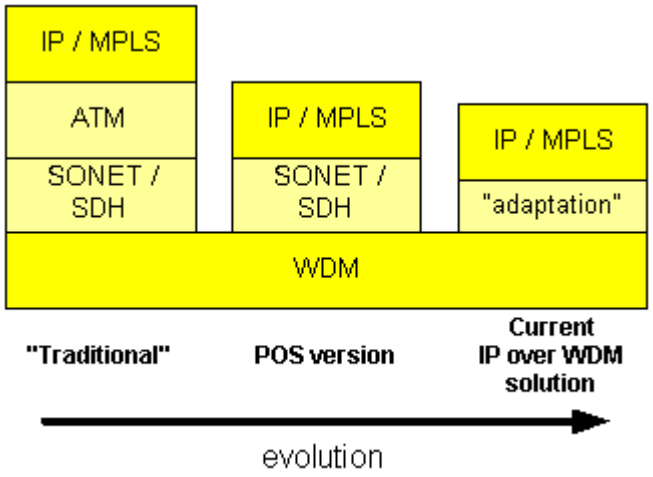


Figure 2: 'IP over optics' evolution

A protocol stack cannot just be replaced overnight. Figure 2 depicts a possible evolution from the current solution towards simpler architectures. The leftmost approach is the full-blown IP/ATM/SDH/WDM solution described previously. ATM can be avoided by using PPP and HDLC framing and in this way simplified architectures can be build [Gha2000]. Because in this scheme packets are encapsulated directly in the SONET / SDH transport modules (or payload envelopes in the SONET terminology) it is called PPP over SONET (POS), which is sometimes referred to as *Packet over SONET*. Going even further a simplified version of the SONET / SDH framing – called adaptation in the figure – can be used to yield what is commonly referred to as IP over WDM. In addition to the framing and encapsulation the adaptation layer provides other functions, such as addressing, access control,

flow control, error checking (including monitoring of the WDM channels) and synchronization. One option for this layer is to use GMPLS.

As the figure suggests, MPLS is being used in most approaches. Mainly, the traffic engineering capabilities of MPLS are very much sought after, because MPLS can then replace the traditional role of ATM and SDH. However, MPLS might be used for even more. Some researchers even envisage MPLS as the technology bringing protection and restoration to the SDH-less architectures [Col2001][Assi2001]. The possibilities of using MPLS for protection/restoration are highly dependent on whether MPLS is overlaid on the optics or an integrated MPLS/optics systems is used. For instance, the label merging facility of MPLS can be utilized to build a scalable mesh of MPLS LSPs on top of the optics and in that way provide fast recovery mechanisms [Col2001].

2.1.1.1. TE and QoS in the Internet

Within the IP community two different approaches for providing QoS in IP based networks have emerged: Intserv or integrated services and Diffserv or differentiated services.

	Characteristics
Intserv	Per flow traffic management. Relies on explicit resource reservation in the network nodes. RSVP [Bra1997] is used as signaling protocol to set up paths and maintain a soft state for each connection, meaning that reservations must be refreshed in order to keep the connection established. All network nodes must support Intserv in order to be able to give any guarantees.[rfc2215]
Diffserv	Per packet traffic management. A number of traffic classes are created by prioritization in all network nodes and classification at network / domain boundaries at the so-called diffserv code-points. The most fundamental diffserv QoS concept is the PHB. Each additional diffserv capable device added to the network enhances the separation of the traffic classes. [rfc2475]

Table 2: Intserv and Diffserv

The main difference between *Intserv* and *Diffserv* is the way resources are treated within the network. The per packet traffic management paradigm used by Diffserv relies on each network element's correct treatment of the packet. The difficult thing here is the mapping from packet processing in each node (so called *per hop behavior* or PHB) to end-to-end connection characteristics. But Diffserv requires no state awareness in the network nodes and hence is very scalable.

Intserv connections are conceptually easier to handle because explicit resource reservation is employed. However, Intserv has a scaling problem. The soft state reservation created by RSVP is accomplished by two types of messages: RESV and PATH. PATH messages flow from the connections initiator and are routed through the network to the receiving client. The receiving client then computes a bandwidth requirement specification and issues a RESV message. This RESV message backtracks the route followed by the PATH message forcing resources to be allocated in the traversed routers. The reason for this cumbersome reservation mechanism is to support multicasting in heterogeneous networks. The PATH massage is just duplicated in branching points and in the reverse direction RESV messages are merged in the branching points.

However, QoS, can not be handled at layer 3 only. The transport protocols (e.g., TCP) can also severely impact the applications' behavior. This is further treated in chapter 6.

2.1.2. IP evolution

IP with its support protocols have driven Internet developments for years. With the introduction of new IP based access networks with potentially a vast number of additional users (e.g., 3G mobile networks [3GPP922] but also more futuristic approaches such as networked coffeemakers and other appliances) there is a actual need for more addresses, but this problem might be solved by proper use of DHCP [rfc2131] and NAT [rfc1631]. This, however, violated the fundamental property that all Internet clients should be equal in terms of connectivity. What about IPv6? IPv6 provides - compared to IPv4, which is the current version – some added features [RFC2460]. The most important being its vastly increased addressing space. In addition, IPv6 provides mobility support with intelligent rerouting functionality and a security framework. However, there are a number of inhibitors to the deployment of IPv6

- Most of the novel functionalities in IPv6 at the time when it came out have now been ported to IPv4 by means of supporting protocols. (e.g., mobile IP [rfc2002]) and new addressing schemes and address translations (NAT) [rfc1631].
- Requires upgrade of *all* the network nodes containing Layer 3 functionality (routers). Most current routers have hardware based, packet forwarding engines [Awe2000] in order to efficiently perform LPM operations, i.e., hardware upgrades are needed to efficiently support IPv6.

These two inhibitors are arguments against an end-to-end protocol. Also at the transport layer PEPs (Performance Enhancing Proxies) that can segment the TCP flow control loop is another argument.

2.1.3. Issues

Protocols are required for the network to work. Protocol stacks must fit the network technologies and thus if the protocol stack is not very flexible then it can inhibit technological developments. The goal must be to find flexible solutions that can make the network evolve.

2.2. Technologies / vertical heterogeneity

This section contains a brief overview of switching schemes of various granularity as well as an overview of wireless technologies.

2.2.1. Packet-, burst- and wavelength switching

Packet switched networks have been around for quite some time [Met96] and it is imperative that the statistical multiplexing property is advantageous. Advances in transmission technology, mainly within the field of optical communication, have vastly increased the bit-rate in networks. An obvious idea is then to combine optics and packet switching to produce all-optical switches and hence eliminating the need for OEO conversion.

The main difference between electrical and optical packet switching is in the data path where the optical packet switch matrix operates on purely optical signals and therefore is capable of switching at very high bit rates [Dan1997][Hun2000][Chi1998]. The optical switches can potentially be fully transparent (with respect to bit rate and payload) but 2R or 3R regeneration is required when cascading several switches [Wol1999].

To resolve contention on the output ports, buffering is required, but the only optical buffering available today is a set of fiber delay lines, which control the delay of the optical packets. Optical buffer space is bulky and hence very limited compared to memory sizes in conventional, electrical switches. In some optical buffer designs the wavelength domain is exploited for contention resolution, thus increasing the effective buffer size. Contention resolution in the optical domain can be done in four ways: wavelength conversion, FDLs, deflection routing and burst segmentation. [Vok2003]

In optical switches, electronic circuits control the packet header analysis and switch-matrix configuration so the switch throughput, measured in packets

per second, is normally limited by packet processing and reconfiguration times. When comparing the properties of electrical and optical switching the following should be emphasized:

- The packet length, measured in bytes, is longer for the optical packet than for the electrical packet, which means that several electrical packets must be bundled into one optical packet. Hence, when mixing electrical and optical packet switches considerations on traffic aggregation are called for.
- Optical buffer sizes are considerably smaller than their electrical counterparts, which mandate shaping of the traffic to avoid buffer overflow.

Optical burst switching (OBS) is a relatively new research area that relies on combining the advantageous traits from optical packet- and circuit- (wavelength) switching. The idea is to first of all use larger "packets" or bursts compared to optical packet switching systems. This clearly relaxes the requirements imposed on the intermediate switches for making forwarding decisions. In addition, the bursts are not packets in the normal understanding of a packet because they are not self identifying. Hence a burst contains no headers that can be used by the intermediate switches. This means that this control information must be made available by other means, but it also significantly relaxes the requirements to the processing capabilities of the switches. Since the packets need not to be processed by the intermediate nodes all packet handling can be carried out all-optically. The required control information on where to send the bursts etc. is provided on a separate channel (wavelength). The control traffic must of course be processed at each node in the network and this is done by OE conversion and electronic processing. Because the bursts are large and the amount of required control traffic low, such a control channel can easily be shared among a high number of traffic channels and might even operate at a rather low bit rate.

Some work has been done on how to do burst-assembly, i.e., how to group the packets that should be sent as one common burst. [Gow2003][Detti2002] This issue is also further treated in chapter 6.

However, there is a number of disadvantages of OBS. Burst grooming / aggregation is not possible in the optical domain and thus OE conversion is required.

2.2.2. Wireless technologies

Wireless technologies have become of major interest during the last 10 years. Within mobile networks, GSM is playing a major role, but EDGE and

UMTS are taking off. Those mobile technologies are access network technologies, but their use might severely impact the core network also. What is particularly interesting is the work on 4G networks, because these are heterogeneous in nature.

2.2.2.1. UMTS

Universal Mobil Telecommunication System is one proposal for a 3G mobile network standardized by 3GPP (CDMA2000 is another for use in the United States). UMTS is based on a Wideband CDMA (WCDMA) access method and compared to the 2G networks offers a number of benefits to the users and operators, among these are increased bandwidth, increased flexibility and support for QoS.

2.2.2.2. 4G – a network tapestry

UMTS may, however, not be enough for future communication needs. Combination of technologies (e.g., satellite, HAP, DVB, DAB. WLAN, UMTS, Bluetooth) is a likely candidate for future, wireless access networks – and is commonly referred to as 4G. Doing this will enable, from a user's point of view, seamless interoperability among the networks. This puts a rather high burden on the terminals that must be able to connect to a multitude of technologies. This is commonly envisaged by using SDR (software defined radio), all the technical difficulties have not yet been overcome.

2.2.3. Issues

Optical packet switching is a hot research topic. However, the research in this area is mainly focusing on the technological issues [Han1998][Chi2003]. The issues of integrating with legacy networks are not widely covered. One problem is that the properties of optical packet switched networks differ substantially from their electrical counterparts. Thus, a requirement for optical packet switching to be deployed in real networks (beside the technological obstacles, which are beyond the scope of this thesis) is that adaptation devices be developed that can interconnect electrical and optical packets switched networks.

In chapter 4 a network architecture that can accommodate such optical technologies and can even catalyze the evolution is presented. The architectures can even be used to accommodate wireless technologies and can thus encompass the entire network. In chapter 6 some modeling and simulation issues are presented.

2.3. Network control

Considerable effort has been put into managing bandwidth in telecommunication networks effectively. One requirement here is a flexible and reliable control system.

2.3.1. MPLS

MPLS (Multi Protocol Label Switching) is a networking concept that is based mainly on a shift of all complex functionality to the edge of the network, leaving only simple tasks for the core network and hence enabling fast and efficient operation. The control plane (that takes care of e.g., routing) and switching (packet forwarding) are completely decoupled, which yields the advantageous property that they can be chosen independently. MPLS is designed as a pure 'everything over everything' concept, hence its name. In reality, however, its predominant use and the majority of standardization work is focused on carrying IP traffic with MPLS, which is due to the ubiquitous Internet.

Packets in MPLS are forwarded along *Label Switched Paths* (LSPs) that are determined by routing protocols based on predefined traffic classes called *Forward Equivalent Classes* (FECs). An FEC can be equivalent to a single entry in a conventional IP routing table or it can be an aggregation of multiple such entries. An FEC can also be specified based on a number of additional constraints such as originating address, receiving port number and QoS parameters. These LSPs are defined in the switches by using labels, which are distributed by a Label Distribution Protocol (LDP) responsible for mapping between routing and switching. The MPLS standard doesn't specify one specific label distribution protocol; it just highlights the required properties. Currently, four protocols are under consideration, of which two are new and two are modifications of existing protocols (BGP and RSVP). [And2000][Rek2000][Jam1999][Bra1997].

2.3.1.1. Label processing in MPLS

In MPLS, switches are generally called Label Switch Routers (LSRs). Ingress edge routers (or more correctly ingress edge LSRs) take care of attaching short, fixed length *labels* to packets when they enter the MPLS domain This includes the non-trivial task of determining to which FEC a given packet belongs. Within the core of the network forwarding will be based on the label only, and before leaving the MPLS domain packets have their label removed by the egress edge LSR (See Figure 3).

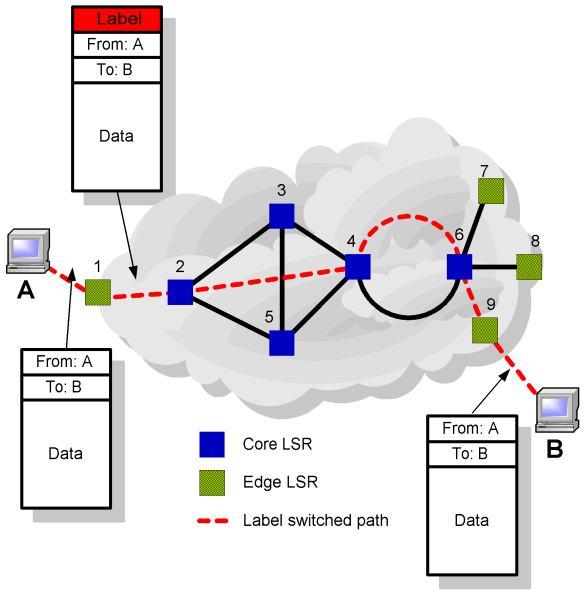


Figure 3: The MPLS label is used only within one domain. By attaching different labels at the ingress LSR, different routes through the network for the same destination can be selected, which allows for traffic engineering.

The labels are generally not kept constant along a LSP and thus a path through the network is defined by a sequence of labels, all of which are assigned by the LDP. In the core switches only the labels are examined. What distinguishes this method from that of conventional IP routing are the loose coupling between the label and the destination address as well as the lookup scheme within the switches themselves. The labels used by MPLS require exact match in the lookup tables, which is a much simpler operation than LPM used for ordinary IP routing, i.e., OSPF would build a routing table is each LSR and based on this information and possibly additional information the label distribution protocol builds another table in (the NHLFE table) which the *label* is used as the key. The outcome of a table lookup is information about outgoing port number and the outgoing label, which is used to replace the label contained within the packet as well as expediting the packet to the designated output port. The label replacement operation is usually called *label swapping* and is the most common packet modification operation in MPLS. In addition, when working with multiple domains in a network, the single label might be replaced by a stack of labels with only the top label being used within one particular domain. At domain boundaries label swapping is insufficient and must be exchanged for more complex operations such as label pushing and popping.

A number of schemes have been devised to simplify this label distribution scheme (e.g., [Gha2000]). In addition, this thesis proposes a new scheme, which is described further in chapter 6.

2.3.2. MPLS assessment

One of the major benefits of the MPLS concept is its ability to perform traffic engineering, i.e., to be able to control how traffic flows through the network, which is one of the prerequisites for providing QoS guarantees on connections. The other major advantage is protocol independence. When wanting to support transport of a new protocol only edge devices need to be upgraded. This feature will make the transition to e.g., IPv6 [RFC2460] smoother.

2.3.3. MPLS extensions and generalizations

Optical packet switching is not yet a mature technology. GMPLS (Generalized MPLS) is a generalization of MPLS that allows a seamless integration of a multitude of technologies, especially circuit switched systems, with MPLS networks. Thus, interfacing traditional telecom TDM systems (e.g., PDH and SONET / SDH) and wavelength routed optical networks is possible with the use of GMPLS.

Optical wavelength switching (or circuit switching) on the other hand is now becoming available and is an attractive alternative in high capacity backbone networks. By using mixed-technology, multi domain networks the advantages of different technologies can be combined. The problem is normally that a unified control and management structure is lacking. However, by integrating MPLS and GMPLS a number of significant advantages are achieved.

As both support traffic engineering, this can be accomplished independently of the underlying technology. What is lacking is a unified control system, which is exactly what GMPLS provides. I.e., the integration of MPLS and GMPLS with these circuit-switched systems is advantageous because is offers:

- Traffic engineering capabilities,
- A high capacity core
- A flexible, controllable edge
- Protocol independence (i.e., e.g., IPvX interoperability)

In Figure 4 a likely usage scenario for GMPLS is depicted– GMPLS is forming 'islands' within an MPLS network. The modeling of such kinds of networks is covered in section 6.

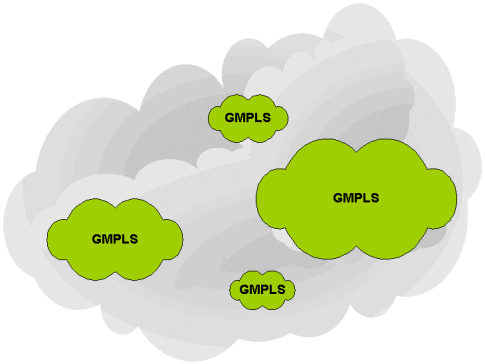


Figure 4: GMPLS in a typical usage scenario where GMPLS is used in some parts of the network

2.3.4. Issues

MPLS is being used for controlling resources in networks. It can be viewed as one large, distributed router, in which the edge routers are the ports and the core routers constitute the internal switching fabric. In chapter 4 a hierarchical network architecture is presented that used MPLS for heterogeneous networks.

2.4. Corporation in an European context

The bulk of the work presented in this thesis has been carried out within EU research projects. This section gives a brief overview of the projects in which I participated and highlights my contributions.

2.4.1. NGNI – NGPN

NGNI (Next Generation Network Infrastructures) is a project started in 2001 with the goal of setting up a forum for sharing information among other project and for creating roadmaps towards future networks [NGNI]. I was involved mainly in the sub-activity NGPN (Next Generation Photonic Networks)

Project objectives

The NGPN sub-activity specifically focused on the use of optical technologies.

Requirements to networks were identified in the deliverable *"Project objectives and benchmark measures for next generation core and metro networks"* [D1] and a roadmap towards future networks was proposed in [D4]. In addition a technology deliverable [D3] and a deliverable with a top-down analysis were issued.

My contributions

My main contribution to this project was the deliverable D4 *"Topologies and architectures for next generation core and metro networks"* [D4], which covered network architectures and protocols as well as migration scenarios.

2.4.2. METEOR

The METEOR (MEtropolitan TErabit Optical Ring) research project ran from the beginning of year 2000 until the end of 2002 [METEOR].

Project objectives

The goal of the project was to design an optical ring with terabit per second capacity. The idea was that such a ring could be used in metropolitan areas.

By using on each fiber 40 wavelengths carrying 40 Gbps each a capacity of a stunning 1,6 terabits per second could be achieved. To connect the ring to the outside world it should be equipped with OADMs in each of its four nodes. In order to keep the optical ADMs simple (and hence cheaper and less bulky) only a subset of the total of 40 wavelengths could be added/dropped in each node. The obviously required careful design considerations because every wavelength must be addable / droppable from at least two nodes.

Within the project a demonstrator of the ring should be built.

My contributions

The applicability of the METEOR ring is not given a priori. One of my tasks was to identify usage scenarios and to come up with suitable network architectures. Part of this work was to identify the need for aggregation devices – 1,6 Tbps is after all a considerable amount of data.

A ring in itself is not a very useful structure, but when it is used to form larger structures its applicability is virtually endless. Hence, architectures in which rings can be combined in various ways are advantageous.

The advantages of using rings are:

- Easy protection. Many protection schemes already exist.

- Simple topology
- Any topology can be decomposed into rings (see Figure 5 below)

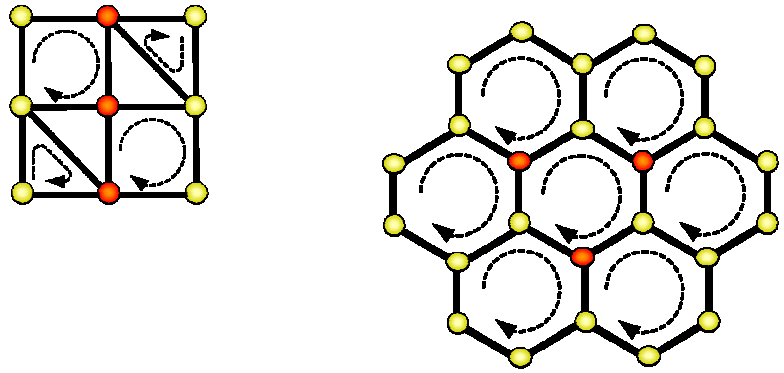


Figure 5: An arbitrary topology can be decomposed into rings. This is one of the reasons why rings are so interesting. The red nodes indicate where bridges between the rings must be implemented. (idea from [Ste1999])

Within the project, resiliency mechanisms were studied and a number of layered architectures of rings and meshed networks were identified and analyzed.

My main contributions are reported in the deliverable D5 *"Network Architecture and System Requirements"*, [MET5]

2.4.3. DAVID

The DAVID (Data And Voice Integration over DWDM) [DAVID] project is an EU funded research project running from the summer of 2000 until autumn 2003. The consortium consisted of 14 partners from universities as well as industry.

Project objectives

The project looked into methods for incorporating optical packet switching into real networks – mainly all-optical core network structures were investigated [Ditt2003][Ditt2001]. There are a number of technological issues in optical packet switching and some of them were addressed within the project. The basics of optical packet switching, however, have been the topic of several other projects (e.g., the KEOPS projects [Gam98]) and hence DAVID tried to identify and explore yet unsolved areas of optical packet switching. DAVID was a very pragmatic approach to the problem and did as such not try to find the definitive solution, but also to find the path towards it. For

instance the use of optics and electronics simultaneously was seen as a optimal way of getting the pros from both worlds.

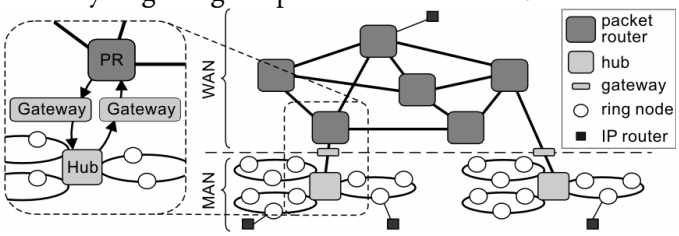


Figure 6: The DAVID network architecture. (from [Ditt2003])

As a proof-of-concept the project also included a demonstrator, which was a small-scale implementation of the concepts investigated.

The DAVID network architecture comprises the MAN as well as the WAN. The MAN part of the network facilitates optical rings interconnected by a hub that ensures contention free traffic exchange. The WAN part is a mesh of high capacity Optical Packet Routers (OPRs). The mesh could for instance be built by using a virtual, light path overlay in a WDM network.

Node architectures. The Optical Packet Routers in the WAN part of the network uses the broadcast and select concept where SOA devices do the switching.

My contributions

The hierarchical network architecture of the DAVID network, this architecture is presented in chapter 4 of this thesis.

Contributions are reporting / published in a number project internal deliverables and in addition in the following published papers / deliverables [p05][p07][p15][p16].

2.5. Summary

By looking at various trends in networking this section has identified a number of focus areas that will form the basis for the discussion throughout the remainder of this thesis.

Optical packet switching is a hot research topic. However, the research in this area is mainly focusing on the technological issues [Han1998][Chi2003]. The issues of integrating with legacy networks are not widely covered. One problem is that the properties of optical packet switched networks differ substantially from their electrical counterparts. Thus, a requirement for optical packet switching to be deployed in real networks (beside the technological

obstacles, which are beyond the scope of this thesis) is that adaptation devices be developed that can interconnect electrical and optical packets switched networks.

Protocol stacks must fit the network technologies and thus if the protocol stack is not very flexible then it can inhibit technological developments.

MPLS is being used for controlling resources in networks. It can be viewed as one large, distributed router, in which the edge routers are the ports and the core routers constitute the internal switching fabric. In chapter 4 a hierarchical network architecture is presented that used MPLS for heterogeneous networks.

3. Network resource administration

"...like playing scrabble with all the vowels missing"

D. Ellington

A network can be viewed as a pool of resources. Resources are temporarily being used when information or traffic is transported through the network or resources are being the reserved. The timeframe during which the resources are in use depends on the protocol layer, the technology and the network administrator's policies. If the network resources are administered well the network's performance is generally high, because that means that the request for resources (the traffic load) matches the resources available. Except for heavily over provisioning the network this is a very difficult problem to solve. A myriad of methods exists for doing network resource administration and depending on the type of network and its use one or more of them might be selected.

Traffic engineering – or the ability to direct traffic to places in the network where capacity is available – is one approach treated intensely in the literature. [Awd2002][Ghan1999][Bane2002] Traffic engineering requires three methods be available in the network. And not only should they be available they must also be coordinated because they are mutually interdependent. These methods are

- Traffic (or utilization) measurements
- Traffic forecasting
- A network control system with traffic control capability

Or in other words one must be able to capture the current network load, determine if this load is desirable or performance (by some metric) could be improved by redirecting traffic and finally a control systems to enforce the decisions about traffic redirections. Doing all this in a coordinated way is obviously very difficult. A locally optimized method for traffic engineering is constraint based routing in which a path through a network is being set up based on a set of constraints.

Resource administration can be done statically or dynamically – depending on the time scale on which resources are being administered. Among the static approaches are network planning, which will be treated subsequently.

The Internet community has come up with very simple approaches such as intserv and diffserv. Of the two schemes diffserv is the only one really being used.

This chapter will cover in some detail the task of network planning. The work presented is based on [p01]

3.1. QoS at different time scales

Quality of Service (QoS) has been discussed as long as networks have existed and the issues related to QoS can still initiate heated discussions at international conferences. One of the reasons is that QoS can be viewed at different time scales:

- Packet level
- Link level
- User perception level
- Network planning level

Depending on the point of view there are different problems and thus different solutions (this might be the reason for the debate!). However, these areas should be jointly considered.

Quality of Service (QoS) is defined by ITU-T as

> *"The collective effect of service performances which determine the degree of satisfaction of a user of a service"* [E.800]

This is a vague definition! It has been widely accepted that QoS should be defined in terms that appear meaningful to the user.

It is difficult to express QoS in absolute numbers. Furthermore, QoS is not necessarily measurable even in theory. To illustrate how difficult a term QoS is, a few examples of "good" (or high) QoS are:

- Flicker free video.
- Telephone service without echo and with an acceptable sound quality.
- Fast reply on remote database queries
- Rapid response on WWW services

These examples show how difficult QoS is to measure. The latter example is obviously related to delay in the network, but how low a delay a user can tolerate before judging a service annoying depends on the user!

In addition to QoS is Network Performance, which is defined as

> *"The ability of a network or network portion to provide the function related to communication between users"* [E.800]

Obviously, network performance and Quality of Service are related. The problem is that the relation is unknown and depends on the actual service offered. Furthermore, poor network performance can still yield satisfactory QoS given that the network provider takes proper actions to compensate for the poor performance (e.g., error correcting codes can compensate for a high BER). The network provider will, however, usually be interested in getting high performance on the network, since this means higher utilization of the (massive) investment in network equipment.

ATM forum QoS, on the contrary, is described by six well-defined, measurable terms, strongly related to network performance, and is one example of an attempt to finding a practical approach to QoS. Examples are peak-to-peak CDV and CLR. ITU-T has included similar terms in their I.371 specification [I.371]. The problem, however, still is, that these parameters are poorly related to QoS at other timescales (and that ATM is receiving still less attention…)

Generally, traffic engineering implies the ability to route along non-shortest paths and utilizes Constraint Based Routing (CBR) where the routes are calculated subject to performance- and administrative constraints, which are assigned by the network management system based on traffic measurements. The well known routing protocol OSPF, which is based on shortest path first (SPF) computations, can rather easily be extended to include constraints, in which case it is called CSPF (Constrained Shortest Path First)[You2003]. This can be done by modifying the Dijkstra algorithm that performs the SPF computations. The CSPF algorithm is "greedy" in the sense that a link is added to a path if and only if it satisfies the constraints, but the route calculations are not globally optimized. (see chapter 3 for further details).

3.2. Hierarchical resource administration

Generally networks are layered and deal with resource administration on a per-layer or set-of-layers basis. Hence application demands cannot be directly mapped to resources for bit transport on the physical medium. Of strong interest is how to build resiliency mechanisms into the network. Resiliency needs special attention on network resources because resiliency costs some-

thing in terms of additional network resources. The subsequent section deals with algorithmic and optimization means for calculating and dimensioning this over provisioning.

An approach for layered networks is presented in [p05] and in chapter 4.

3.3. Network planning issues

The solution could be to introduce long-term traffic engineering based on e.g., Linear Programming (LP) techniques that will produce more optimized route assignments compared to constraint based routing. LP is very time consuming, however, so a combination of LP and CSPF is preferable, i.e., while CSPF calculates the routes in the network, an LP solver runs in the background and optimizes the CSPF computed routes. Another approach to reducing the LP computation time is by employing heuristics that, albeit giving only sub-optimal solutions, are usually close to the optimal ones with significantly reduced computation times. [Pio2000]

Generally, using capacity in a network is associated with a cost. If the traffic matrix (i.e., the set of traffic demands between all source/destination pairs) is given one can lay out the network efficiently by solving an optimization problem, namely that of minimizing the cost of carrying the given traffic – subject to given demands for protection. In the following section an example of how to use linear programs for capacity planning in networks is given. Optimization by LP or IP (Integer Programming) uses the branch and bound approach to search the solution space for the optimal solution.

3.3.1. An example optimization problem

Optimal network design is an interesting case to consider. This section will show by example how Linear Programming can be used to do network planning/optimization.

Consider a network of arbitrary topology and a set of demands specifying the traffic matrix for the network. The task is now to assign capacity to each link such that the traffic demands be satisfied and at the same time minimizing the total cost. Below is given the notation used in the linear programming (LP) formulation of the optimization problem. In addition to considering the network when everything works as specified it also takes into account link failures. In this way the network is designed to cope with failures, which is the starting point for making a resilient network. Still the issue of detecting the failures and acting upon them, i.e., network control and management, remains. Failure situations are indicated by the s indices. The nominal - or failure free - situation is indicated by s=0. A total of J predefined paths are

given as input to the optimization as well as the demands. The network topology is given as a set of binary variables (a_{edj}). It is important to stress that this will work for an arbitrary network topology, i.e., the variables a_{edj} define the topology (see also [p01]).

$e \in E$	Link indices
$d \in D$	Demand indices
$j \in J$	Path indices
$s \in S$	Failure situations.
c_e	Cost of link e
y_e	Capacity of link e
b_d	Path diversity coefficient for demand d
h_{ds}	Volume of demand d in situation s
x_{djs}	Flow, realizing demand d on path j in situation s
M	Capacity module size
a_{edj}	$\begin{cases} 1, \text{if path j for demand d is using link e} \\ 0, \text{otherwise} \end{cases}$
α_{es}	$\begin{cases} 1, \text{if link e fails in situation s} \\ 0, \text{otherwise} \end{cases}$
δ_{djs}	$\begin{cases} 0, \text{if a link on path j for demand d has failed} \\ 1, \text{otherwise} \end{cases}$

A formulation of the nominal design case suitable for solving by using CPLEX is given below:

Objective:

$$\text{Minimize} \sum_{e \in E} c_e \cdot y_e$$

Subject to:

$$\sum_{j \in J} x_{dj} = h_d \qquad \forall d \in D$$

$$\sum_{d \in D} \sum_{j \in J} a_{edj} \cdot x_{dj} \leq M \cdot y_e \qquad \forall e \in E$$

To add path diversity an extra constraint must be added to the problem to ensure that not all capacity of a demand is realized by using only one path. The constraint is formulated by using a path diversity constant b, where $0 \leq b \leq 1$. In order to ensure a problem, which has a feasible solution, b must be calculated considering the actual number of paths for a demand. It doesn't make sense to require that a demand be split on for instance 3 paths if there are only 2 available. I.e., b_d denotes the path diversity constant for demand d and is given as:

$$b_d = \max(b, \frac{1}{|J|})$$

With this definition of b_d the problem is cast into the following formulation:

Objective:

$$\text{Minimize} \sum_{e \in E} c_e \cdot y_e$$

Subject to:

$$\sum_{j \in J} x_{dj} = h_d \qquad \forall d \in D$$

$$x_{dj} \leq \lceil b_d h_d \rceil \qquad \forall d \in D, j \in J$$

$$\sum_{d \in D} \sum_{j \in J} a_{edj} \cdot x_{dj} \leq M \cdot y_e \qquad \forall e \in E$$

Either flows or links can be protected. Here only restricted reallocation in the case of flow protection is considered. In the following problems path diversity is not considered.

Failures are described by introducing the notion of situations. A situation is a distinct scenario, in which a given set of all links fails. However, in the cases considered here only single link failures are taken into account. This is a rather realistic situation thou, since the probability of two simultaneous fiber cuts is small.

When adding protection to a given network one has to consider:

1. The amount of nominal capacity released due to the failure situation.
2. The spare capacity required for reallocating the "broken" flows.

The goal is to minimize the amount of excess (protection) capacity.

Objective:

$$\text{Minimize} \sum_{e \in E} c_e \cdot yp_e$$

Subject to:

$$\sum_{j \in J} x_{djs} \geq h_{ds} \qquad \forall d \in D, s \in S$$

$$x_{djs} \geq \delta_{djs} x_{dj0} \qquad \forall d \in D, j \in J, s \in S$$

$$\sum_{d \in D} \sum_{j \in J} a_{edj} \cdot x_{djs} \leq M \cdot \alpha_{es} \cdot (yp_e + y_e) \qquad \forall e \in E, s \in S$$

In this case the network resources should be allocated in such a way that failure situations could be coped with without affecting resources allocated in the nominal state. I.e., in case of a link failure some flows will obviously be affected and thus resources for those flow must be reallocated. However, flows that are not affected by the link failure should be preserved (restricted reallocation).

Objective:

$$\text{Minimize} \sum_{e \in E} c_e \cdot y_e$$

Subject to:

$$\sum_{j \in J} x_{djs} \geq h_{ds} \qquad \forall d \in D, s \in S$$

$$x_{djs} \geq \delta_{djs} x_{dj0} \qquad \forall d \in D, j \in J, s \in S$$

$$\sum_{d \in D} \sum_{j \in J} a_{edj} \cdot x_{djs} \leq M \cdot \alpha_{es} \cdot y_e \qquad \forall e \in E, s \in S$$

Where δ_{djs} is used to determine whether a flow in a given failure situation is affected or not.

These formulations are linear programs, which can be solved by using an LP solver such as CPLEX [CP99]. The formulation above utilizes path protection, i.e., in case of a link failure rerouting the affected paths restores all paths from source till destination. Another basic protection mechanism is link protection. In this case protection is carried out on a per link basis, i.e., the complete path for all connection is kept except for the failed link where paths are found to replace the failed link. Generally, link protection is much more costly than path protection. However, it is much more simple because it can be carried out locally opposed to path protection that requires full topology knowledge. Selecting one of these approaches is really a cost / complexity tradeoff.

A basic requirement for either of these schemes to work is that the network be designed in such a way that disjoint paths exist between all source/destination pairs and in addition to that the network routing must take these paths into account. A well-known algorithm for computing sets of disjoint paths is Suurballe's algorithm [Suu1984].

When considering layered architectures it is very important to make sure that resources are physically disjoint. This can be done e.g., by using Shared Risk Link Groups (SRLGs). [Elli2003]

3.3.2. Why is optimization so difficult?

This is a simple example to illustrate that even simple problems can have quite complex solutions. If an apparently optimal solution could be optimized even further that would clearly be great. One way of doing that is by

allowing the splitting of flows. Generally it turns out, that by splitting flows onto multiple links a much more effective use of resources can be achieved (flow is bifurcated). However, taking this into account severely complicates the optimization problem. Figure 7 illustrates this by an example. A total capacity of 3 (arbitrary capacity unit, e.g., wavelengths) must be allocated between the two yellow nodes and there must be build-in protection such that any of the three links can be cut while maintaining a reserved capacity of three. In figure a) the entire flow must be carried on one single link. Hence a backup link with the same capacity is needed and the total capacity required is 6. If splitting of the capacity is allowed (as in figure b) the total required capacity could be reduced to 5, while maintaining the same degree of protection. The best utilization occurs in the situation where non-integer flows are allowed (figure c) in which case only a capacity of 4.5 is required. It should be noted that an optimization program would not find such a solution because splitting of flows is not considered.

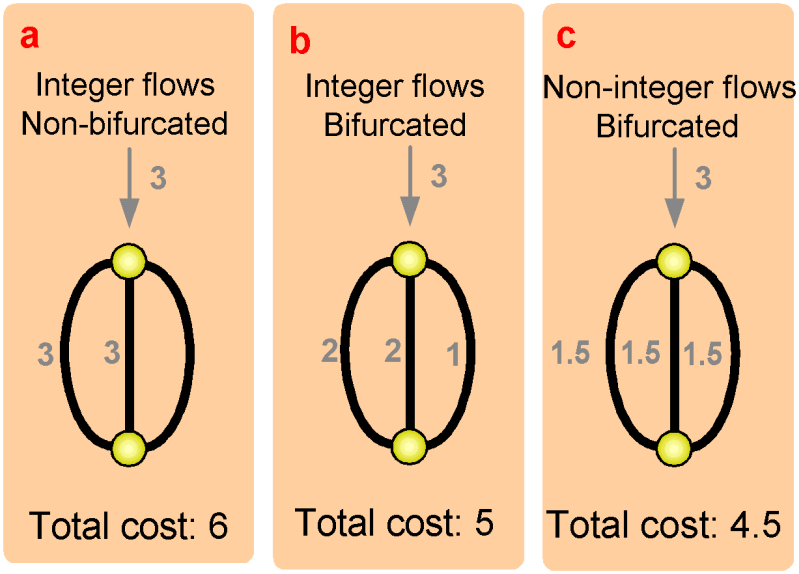


Figure 7: Example: When splitting flows a more optimal use of resources can be obtained.

Linear Programming is very time consuming and for large and / or complex problems (e.g., when considering multi layer optimization, i.e., optimizations of several interdependent layers simultaneously) one cannot be sure of finding an optimal solution within a reasonable timeframe (or at all!) because the problems are NP complete (because it is actually just another formulation of a problem known as the multicommodity flow problem).

3.3.3. Assessment

Optimization by linear programming (LP) has clear advantages and drawbacks. These are highlighted below

Pros

- The network capacity can be used efficiently. This is of course a desirable property and is the main reason why it is used. A network operator wants to get money back for the massive investment of deploying a network.

Cons

- LP is very time consuming. Thus it can only be applied to network planning not to (constraint based) network routing and restoration. To resolve this heuristics could be used. [Pio00] When using heuristics the network planning process can be carried out so fast that it can be used while the network is operating. I.e., once in a while one can by measurement find the actual traffic demands to the network. Based on these measurements the heuristics can be used to find the (almost) optimal routing of the traffic. If desired the traffic can then be rerouted.

For that reason approximations are employed, e.g., heuristics and evolutionary algorithms. These are methods for quickly finding good (but suboptimal) solutions to optimization problems.

3.4. Heuristics

Simulated annealing (SAN) is a widely known heuristic that works with full allocation stated as representation of solutions. The optimization is controlled by a variable known as the temperature, which determines how often a worse solution should be temporarily accepted in the search for a global minimum. The temperature is slowly decreasing during the process (hence the name simulated annealing)

Simulated Allocation (SAL) was proposed recently by Pioro [Pio97] and is a meta heuristic. Contrary to SAN it works with also partial allocation states in its search for a solution. As was demonstrated in [p01] the simulated Allocation scheme is superior in terms of required time to the traditional simulated annealing approach.

3.5. Summary

In this chapter some considerations on network resource administration has been presented. Resource administration of some kind is a requirement for QoS guarantees. Looking at QoS on a large time scale requires setting up the requirements for the network capacities, i.e., it is the process of network planning. Linear programming has been shown as one way of doing network planning and it was shown how spare capacity for resiliency can be taken into account. However, LP is very time consuming, which limits its applicability. Heuristics can shorten the time considerably at the expense of a less optimal solution.

4. Network architectures

"If there is nothing new to be found in melody then we must seek novelty in harmony"

G. Ph. Telemann

In the old days, the vision was to develop one single technology for multi service networks. This was one of the drivers behind developing and deploying ATM. However, the technologies being developed today are of a different nature. It is no longer likely with one single technology, simply because the vast amount of equipment in e.g., the global Internet makes instant upgrade/replacement impossible. I.e., gradual upgrade of networks creates heterogeneous networks consisting of a number of different technologies. Now, for instance, optical technologies are being introduced into the core networks, but electrical routers/switches are still present and must coexist with the new technologies. Thus, the networks of the future will be multi technology, multi service networks. In order to make such network useful new network architectures that can embrace the diverse nature of networks must be developed. Is that different from the current Internet? Yes! The Internet Architectures uses IP to homogenize a set of diverse, interconnected networks.

In this chapter hierarchical network architectures are investigated and a novel, hierarchical network architecture primarily targeted heterogeneous networks is introduced.

4.1. What is a network architecture?

A network's architecture defines its overall structure in terms of physical and logical entities and their mutual relationships. Thus the term network architecture comprises physical network elements, their interconnections (i.e., the network topology and the transmission technology being used) as well as the protocols on top that govern how information is exchanged. Network architectures can be subdivided into two categories – logical and physical – these are treated in the subsections below.

4.2. Logical network architectures

A logical network architecture is as its name implies a logical structure completely decoupled from the physical network. Logical topologies can be built at any level even at the transport or at the application level. They can be static or dynamically reconfigurable.

4.2.1. Examples

Peer to peer networks (P2P networks) have emerged recently and are examples of application level logical network architectures [Rip2002]. In such networks clients automatically detect network connectivity between members of the P2P network and establish their own view of the world, i.e., they create a logical overlay to the physical topology. This logical topology is now used as a basis for routing, which can lead to a large degree of inefficiency.

Two examples are shown in Figure 8. Figure 8 a) depicts a logical topology which matches the physical one, whereas in Figure 8b there is a mismatch. In example b) broadcast from the blue node would require 6 times the capacity on the blue link compared to example a). This is due to the fact that the peer-to-peer protocol has a view of the network, which differs radically from the network's physical topology.

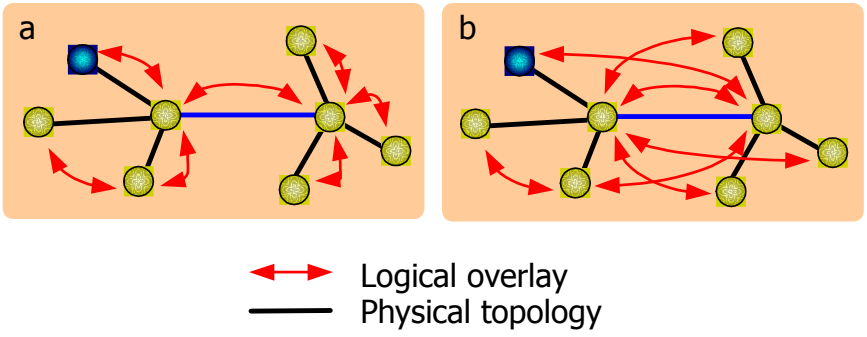


Figure 8: Logical versus physical architecture (idea from [Rip2002])

Another example is the use of BGP in the Internet. BGP creates a routing overlay, based on the ISPs' preferences and policies, which might differ substantially form the 'best' route seen from a physical topology point of view.

4.3. Physical network architectures

Physical architectures are defined by the physical properties of the network. One example is an optical network with a WDM overlay.

The figure (Figure 9) below shows how, physically a node can have properties that makes it belong to several technologies, e.g., WDM, and optical packets switching [MET5][p07]. Such a node can handle resources with varying granularity, e.g., fiber, wavelength and optical packet / burst.

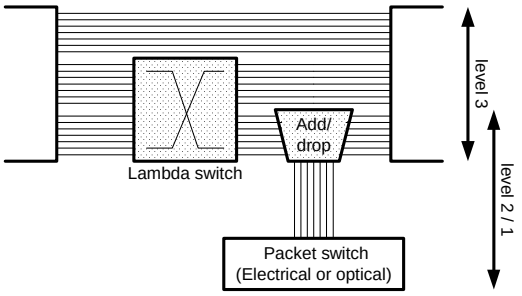


Figure 9: network node structure [p07]

4.4. Mixed technology networks

A consequence of migrating from one, existing network architecture to another one is that new equipment must be introduced. The sheer size of networks often dictates a step-by step migration strategy, which implies that at all times the network will consist of a mixture of equipment, ranging from e.g., electrical routers to all-optical packet and wavelength switches. It is important to find a suitable architecture in which a new technology (e.g., all-optical switches) can be introduced gradually and hence enable a seamless migration. It should also be emphasized that this mixed technology network, which for instance could be organized in a hierarchical fashion (see section 4.4.2), is in fact advantageous for many networks because it makes the network operator leverage a number of different technologies while seamlessly migrating to the newest technologies. In addition, it can be exploited when doing resource reservation /administration.

4.4.1. Single- versus multi technology architectures

A network can be single technology/protocols in many levels. ..e.g., IP level. A network architecture based upon one single technology has the advantage of easier maintenance; this is one of the main arguments for the widespread use of IP, for instance. However, a single technology probably cannot provide the optimal solution in all circumstances. In any case, it is a given fact

that most of today's telecommunications networks are multi technology networks. There are three main reasons for that:

1. Interconnection.
   The networks are themselves interconnected networks of different operators. Each operator has full control over its own domain (and adopt a single technology, if possible), but has no control over what technology its neighbor is using.
2. Size.
   Some networks are very large. Each technology has its advantages. Therefore different technologies can be optimal in different circumstances/environments, and thus an operator can decide to use the optimal technology solutions in the different areas.
3. Upgrade.
   The upgrade of large networks is done gradually and during the upgrade phase, multiple technologies are present in the network. This can also occur do to competition in standardization.

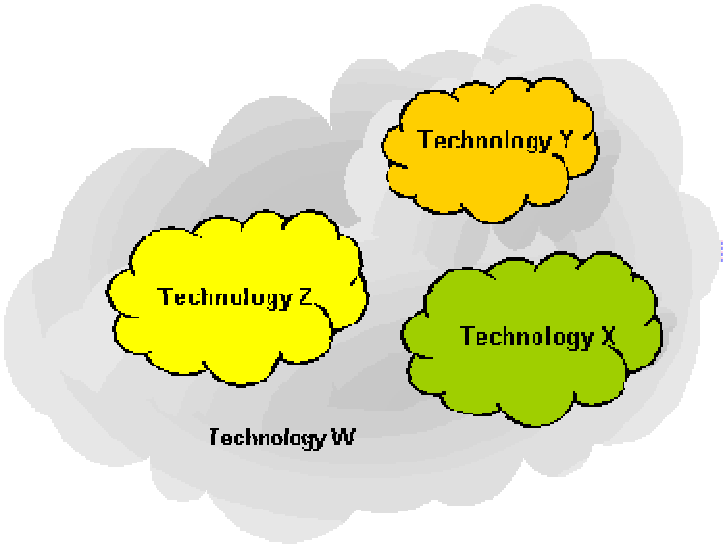


Figure 10: A mixed-technology network. Due to network evolution new technologies are popping up as "islands" within the network.

In multi-technology networks it is straightforward to group all network nodes into areas or domains such that within each domain there is only single-technology equipment. This is depicted in Figure 11: Structure of a

heterogenous network. Between network domains are gateways that work as adaptation devices.

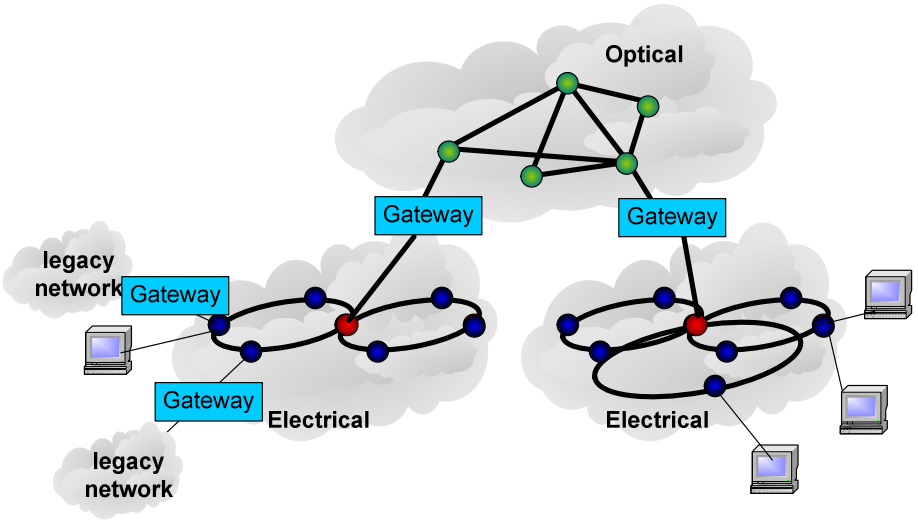


Figure 11: Structure of a heterogenous network. Between network domains are gateways that work as adaptation devices

With this partitioning of the network one must find a way to interconnect the network domains so that the entire network remains fully connected. There is then a need to do traffic adaptation between the areas for the following reasons:

- *Bit rate difference:* Different technologies support different bit rates. At the network edge there could be adaptation from e.g., Ethernet (e.g., 100 Mbps or 1 Gbps) bit rates to the standard telecommunication bit rates (622 Mbps, 2.5 Gbps etc.). In the core network the adaptation will usually be required when the bit rate changes by a factor of four. (e.g., from electrical switches running 2.5 Gbps to optical switches running 10 Gbps or 40 Gbps)

- *Packet size variation:* Usually the switching speed is packet-per-second limited. I.e., when going to a domain with higher bit rate the timely duration of the packets is unchanged and thus the size of the packets, in terms of bits, increases with the bit rate. One example is two optical domains using the same switching technology but with bit rate difference. Usually the bit rate increases by a factor of four and thus in this case the packets in one domain will contain four times as many bits. Another example is where the bit rate is kept constant but the timely duration is increased. The technologies used, e.g., packet switching and fast circuit switching can mandate such a duration increase.

- *Packet length constraints:* The domains might use fixed or variable length packets
- *Transport characteristics:* Various technologies will most likely possess different transport characteristics with regard to e.g., packet delay / loss.
- *Traffic grooming:* When entering a new area it might be advantageous to merge traffic streams. Two or more streams with same destination network can be combined and hence make more efficient use of network resources.
- *Addressing schemes.* Network administrators usually administer their own pool of addresses. Hence for global interconnection either administration or translation at domain boundaries is required.

One important conclusion that can be made from the above list is that even in the case where a adaptation unit serves as an interface between two network domains running at the same bit rate a traffic adaptation device / gateway might be beneficial, mainly due to the need for traffic grooming.

When doing traffic aggregation / grooming there's one obvious trade-off namely that of delay versus bandwidth utilization. When aggregating packets one can choose to optimize for utilization/efficiency *or* delay. Obviously, if outgoing packets are filled up completely this is an optimal usage of bandwidth resources; however, the price to be paid for this efficiency is increased delay. On the other hand the delay could be minimized at the expense of poorer bandwidth utilization. (see section 6)

4.4.2. A hierarchical, mixed technology network

Obviously, logical and physical network architectures can be mixed. Figure 12 depicts a hierarchical network architecture comprising at the two lowermost levels a hierarchical, physical topology, on top of which a logical architecture is built. This logical architecture is itself hierarchical. The logical layering on top of the physical topology can be exploited by e.g., hierarchical routing protocols. (e.g., PNNI [PNNI2002], which is the most prominent example, but a number of hierarchical routing protocols for ad-hoc networks exist also.[Mie99][Roy99])

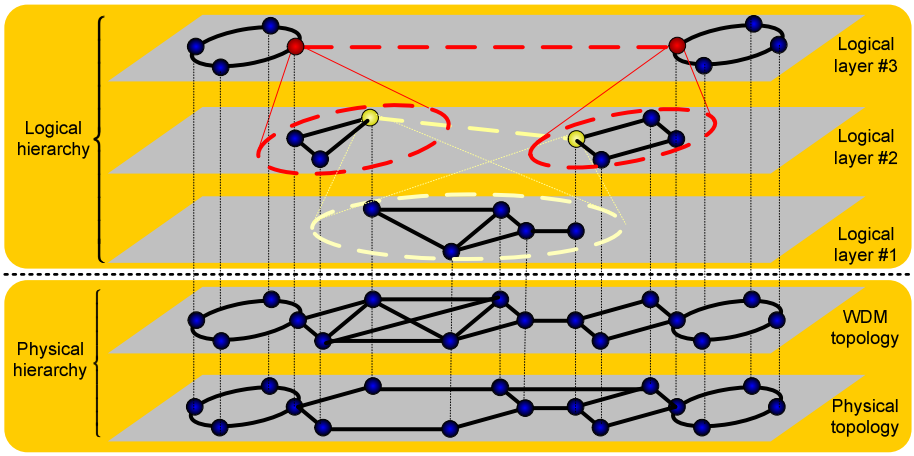


Figure 12: An example hierarchical network

Routing and label distribution is performed within each level of hierarchy. Each LSR participates in routing, including the optical label switches at Level 2. Generally, routing information from Level N+1 is aggregated and distributed to Level N. That is, all Level N+1 routers will be presented as a simplified topology towards Level N. Figure 13 depicts some possible representations of aggregated topologies. The various representations differ in complexity and thus require a different amount of information to describe them. When topologies are aggregated, information is lost. Hence, the information available to the routing protocols is reduced, which yield less optimal routes. Thus, the choice of the way to represent an aggregated topology is a trade-off between the amount of routing information and accuracy in routing computation. Aggregated topologies are also shown in Figure 12. The dashed lines covering a group of nodes at one level is represented as a single, logical node at the level above it.

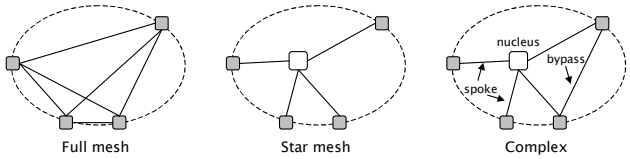


Figure 13: Three possible ways of representing aggregated topologies (Based on [PNNI2002])

During the routing process, information from Level N is used as input to the routing decision process at Level N+1. Route calculations are performed within Level N+1 and an aggregated routing table is computed. The aggregated routing table contains translation between address prefixes and destinations at Level N+1 (A destination at Level N+1 is a LSR with Level N connec-

tivity). Label distribution can be performed within each level following the principles of distribution in a flat network.

4.4.3. Granularity in resource administration

The hierarchical architecture reflects the technologies and their granularity in resource reservation. [p03]

4.5. Hierarchical MPLS (H-MPLS)

Figure 14 is a top down view on the network from Figure 12, in which the different levels of hierarchy have been collapsed into a flat network structure.

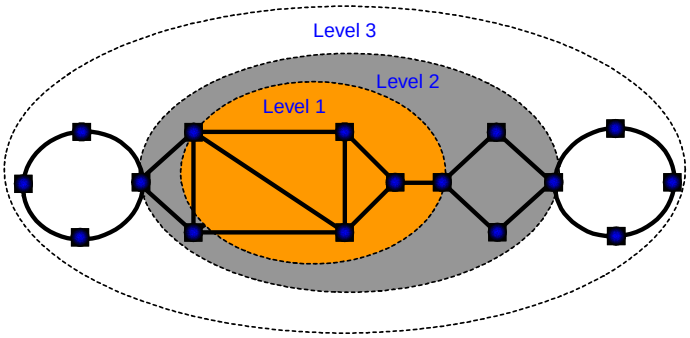


Figure 14: collapsed hierarchy

Each hierarchical level, which then becomes an area in the flat network, is denoted *'level N'*. The different levels (1,2 and 3) within Figure 14 could also be called *Electrical MPLS* (EMPLS), *Optical MPLS* (OMPLS) and *MPλS*, respectively. The subdivision is performed based on device characteristics and these names emphasize those characteristics. To keep the hierarchical structure in mind, the area names refer to levels. This scheme can easily be extended to also cover the access network. This could be called hierarchical MPLS and is based on the work done within the framework of the European Research Project DAVID ([p05], [p07], [p09]).

The following two examples illustrate how this works.

4.5.1. Example – core network

This section shows how the concepts of MPLS can be utilized to create a unified switching /routing approach covering the entire network comprising as well electrical as optical packet switches. The Electrical MPLS level is composed of packet routers that perform switching of packets electrically.

Switching of EMPLS packets consists of swapping the label in the electrical packet header: a forwarding table specifies the mapping of (incoming interface, incoming label)-pairs to (outgoing interface, outgoing label)-pairs. The EMPLS level is used at the periphery of the hierarchical network, where the capacity of the electronic routers is sufficient. Functionality that requires a large amount of buffering, such as advanced scheduling schemes and flow merging as well as MPLS edge functionality, can be conveniently implemented at this level. Furthermore, the EMPLS domain should take care of conditioning the traffic for the optical network.

At the optical packet level (OMPLS), the packet payload is switched transparently within the optical packet routers in order to handle higher throughput nodes (higher bit rates and more wavelengths). Mappings from (incoming fiber/wavelength, incoming optical label)-pairs to (outgoing fiber/wavelength, outgoing optical label)-pairs are based on a forwarding table.

The OMPLS level utilizes optical packet forwarding to achieve very high throughput. The high capacity/throughput OMPLS level is used only where such throughputs are required and where the traffic have already been aggregated and conditioned properly. Finally, within the OMPLS level, traffic can then be aggregated even further so that eventually an entire wavelength worth of traffic is collected. If that is the case, it might be beneficial to add an additional, lambda-switched level that utilizes MPλS [Gha2000].

The MPλS level equipment switches at the wavelength level and as such is a wavelength-routed network without a virtual packet overlay network. That is, the wavelength indicates the destination, and the wavelength can be interpreted as a label at that level. Within nodes belonging to the MPλS level, wavelength assignment is performed by MPLS control plane functions. This approach can be extended even further. In line with the Generalized MPLS (GMPLS) one can also perform switching on fibers, which could then adequately constitute the uppermost level in the network [Man2001].

Label switches running at 10 Gbit/s interface speed are feasible in electronics, but 40 Gbit/s operation will probably require optical packet switching where the payload bits can be switched transparently. Optical versus electrical switching at 10 Gbit/s is a matter of equipment cost, flexibility, and required throughput in the switch. The bit rates of level 2 and 3 are identical, because it is very difficult to change the bit rate of purely optical signals, mainly because that for aggregation, buffering is inevitable, and optical buffers are of limited size.

4.5.2. Example – access network

The figure below depicts an example, layered architecture being used in the access network. It shows how a node can belong to several layers / levels simultaneously, this could be e.g., a 4G network. The technologies shown in the figure are quite different from the core network technologies treated previously.

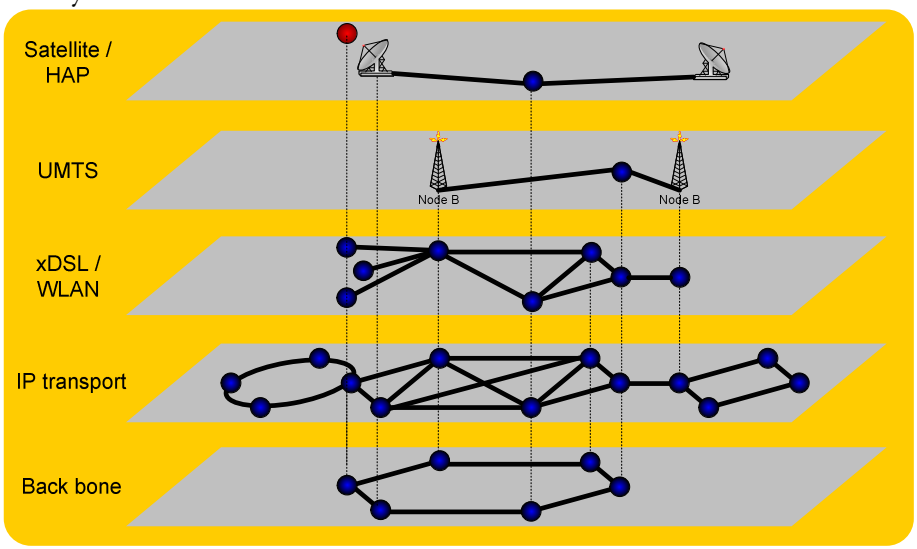


Figure 15: Hierarchical network architecture being used in the access network.

In this hierarchy there is probably no need for packet aggregation because rather small packets are used everywhere, but still between levels one could benefit from grooming of traffic. The MPLS approach to this heterogeneous network will act as a common control platform.

The *"back bone"* part of Figure 15 could then be a H-MPLS core network as described in the previous section.

4.5.3. Hierarchical MPLS label operations

The hierarchical structure of the network is reflected in the MPLS label stack. Within each level of hierarchy, the label is swapped in the LSRs. A new label is pushed on the label stack when the packet leaves Level N and enters Level N+1, and when the packet re-enters level N the label is popped off. Thus, a packet at Level N has at least N entries in its label stack. If aggregation is used between the levels then there is generally more than N labels attached to the packet.

As described above, each entry in the label stack corresponds to a level in the network hierarchy. It is therefore possible to aggregate Level N traffic streams (LSPs) going to the same destination in Level N+1, i.e., a number of Level N packets, with identical Level N+1 destination, can be sent together in a larger, composite level N+1 packet.

4.5.4. Gateway functionality

This section look into the functionality required in the gateway devices and how this fits within the MPLS concept.

In the following, only fixed length packets are considered. The packet length may however vary among the different levels of hierarchy, i.e., the packet lengths are fixed within a level and converted at level boundaries. It is assumed that switching at each level of hierarchy is limited to a certain number of packets per second. Thus, the packet duration is almost identical at each level, but the size in bits increases proportional to the link bit rate.

The boundary between hierarchy Level N-1 and Level N is located at the interface of the Level N label switching router. This means that links between Level N-1 and Level N routers belong to Level N-1, and the packet format at this interface link is that of Level N-1. I.e., it is possible to connect a number (e.g., 4) of Level N-1 links to a common Level N interface.

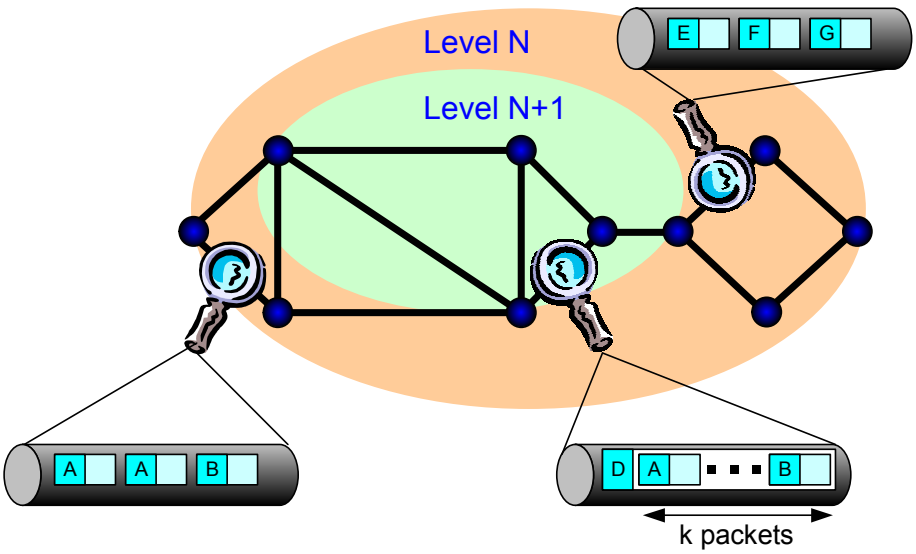


Figure 16: Flattened network

Traffic aggregation and label stacking are depicted in Figure 16, where an example (flattened) hierarchy with only two levels is shown.

Consider the example in which the LSPs (Label Switched Paths) of the A- and B-labeled packets in the figure have identical destinations within Level N+1. A number of these packets can then be aggregated and, according to the MPLS concept of label stacking [rfc3032], the D-label is pushed onto the label stack. The D-label is popped off the stack when the Level N+1 packet reaches an egress Level N+1 LSR, and the Level N labels are swapped into E. F and G respectively. The final Level N destinations are determined from the E, F and G labels.

Note that the egress LSR at Level N+1 must perform label swapping of Level N packets. This is a requirement for topology aggregation, where the entire Level N+1 network appears as a simplified Level N LSR.

The functional parts of the aggregation unit are shown in Figure 17. M Level N flows are merged into one Level N+1 flow. The M label processing units determine the Level N+1 FEC that indicates a specific FEC queue. The scheduler selects a particular FEC queue for transmission, and up to k Level N packets are taken from that queue and sent together in a larger Level N+1 packet. Typically, k is greater than or equal to M in order to make the aggregation scheme work conserving. The total number of queues depends on the size of the network and the number of QoS classes defined for the network.

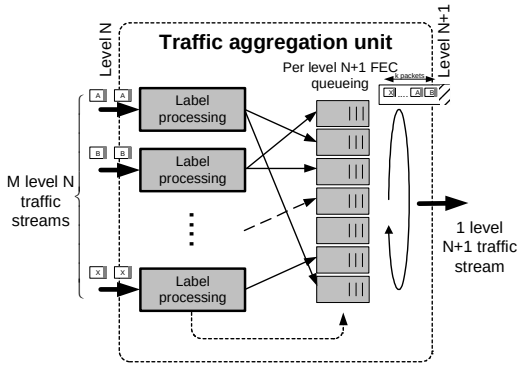


Figure 17: In the aggregation unit M traffic streams are aggregated subject to their characteristics. At level N+1 the bit rate is k times that of level N.

4.5.5. Implications and applications

The partitioning into FEC queues is related to the QoS approach for the network. Prioritization based on the DiffServ concept sub-divides the traffic into a number of distinct service classes. The packets in a particular queue must belong to the same FEC and have identical Level N+1 destination, which yields a total of Q * D queues, where Q is the number of service classes (typi-

cally smaller than the number of FECs) and D is the number of *Level N* destinations.

A Level N packet is selected for transmission by a scheduling algorithm. The scheduling algorithm ensures an efficient aggregation while keeping the delay bounded. The scheduling decision can be performed hierarchically; first among service classes and then among Level N destinations within the selected service class. The result is a more scalable solution compared to a non-hierarchical scheduler, i.e., a complexity of O(Q+D) instead of O(Q*D). The scheduler and its impact on network performance is treated further in chapter 6 and in [Ber2002].

Traffic aggregation has its impact on the statistical traffic distribution across the network. It is expected that a statistical multiplexing benefit is achievable, which is also very desirable since the buffering resources decrease for the higher levels of hierarchy (i.e., small fiber delay line (FDL) buffers compared to large electronic buffers).

4.6. Requirements to future networks

It is hard to set up the requirements to future network architectures as it to a very large extent depends on the requirements imposed on the network – which is unpredictable. The volume of traffic has increased and its characteristics have changed substantially. One reason is due to peer-to-peer applications [Sub2004]. However, exactly this uncertainty imposes one very important requirement; *flexibility*.

Flexibility is important when selecting technologies, when designing the architectures and when operating the networks. The future networks will most likely be multi-technology networks due to the sheer size of backbone networks. Thus, decisions on how to design, deploy and manage such multi-technology and at the same time multi-service networks must be made. Instead of striving to find one common solution a more pragmatic approach is to create an architecture, which allows the diversity to blossom.

4.7. Summary

A number of requirements to future networks can be set up:

- High capacity
- Transparency (signal format, protocol independence)
- Traffic engineering capability
- End-to-end QoS support

- Flexibility

All of these requirements can be met by using a combination of technologies along with hierarchical MPLS – or H-MPLS, which is a novel scheme for combining all the networks and make them seamlessly interoperate. This might for instance be an excellent way of harnessing the power of optics. Such architecture might be beneficial for heterogeneous networks where technologies with diverse characteristics must be accommodated.

- Optical networks have huge capacity but consideration must be made on how to exploit this enormous capacity.
- Electrical, wired networks offer moderate capacity and higher flexibility.
- Wireless networks offer low capacity but great flexibility in terms of user mobility – this is particularly true for the mobile, wireless access networks.

The important thing here is the gateway device that takes care of adaptations and thus enables interconnection of the various levels. Such devices and how to model them will be treated further in chapter 6.

5. Performance evaluation by simulation

"Things should be as simple as possible, but not simpler!"

A. Einstein

Modeling refers to the - usually computerized - imitation (and not replication!) of a real-life system or subsystem. Model development relies on a set of assumptions and is an art rather than a science! Model consistency / credibility is ensured through validation and verification and questions about the real-life system can now be answered by performing experiments with the model, thus artificially generating the history of a system. The process of doing experiments with such a virtual world - a computer model - is called simulation and is an ideal tool for performance evaluation [Jai91].

This chapter gives an overview of modeling in general and of how to model networks, specifically. Models should be as simple as possible in order to ensure that development, verification, validation and simulation can be carried out with a reasonable timeframe. The concept of mixed complexity modeling is introduced as a method for devising simple models. At the end an example of how to model a satellite network is given.

5.1. Performance evaluation by simulation

Performance is a key criterion in a number of cases. Thus, doing performance evaluation is a prerequisite for knowing the system's performance. Performance evaluation requires at least two tools: a *load generator* to apply input to the model and a *monitor* to measure the results.

Modeling is useful for performance evaluation in a number of cases; here is a non-exhaustive list:

- Benchmark systems that do not yet exist.

- Evaluate very large and complex systems.
- Evaluate systems that are not available for measurements
- Perform what-if scenarios

What is a model? A model is an imitation of some predefined features of a given system. Examples of real life systems are a nuclear power plant, a chemical process, a packet-scheduling device in a network etc. Obviously the modeling might be done a little bit differently for these systems, but there are similarities.

This chapter will focus on performance evaluation studies that can be efficiently carried out by using event driven simulators.

5.1.1. Modeling methodology

Models are descriptions of systems. The real-life system must be simplified greatly in order to be able to build a model that can produce results within an acceptable timeframe. It can be very tempting to just model everything and then use the model to gain understanding. However, such a brute-force modeling methodology that just tries to model the real network in every detail is often inappropriate. Below the goals for the simulation are identified and based on that the simplified simulation model can be set up. Obviously, the model must be simple enough to achieve the identified goals, while representing a fair model of the real network.

The tasks of the simulation process

Naturally, the system and the way it works have to be described. Each building block of the whole system may be described with a variable level of detail, which depends first on the precision the study is to provide (think of a system involving communication between sub-systems: the description may choose to incorporate the details of the communication protocol, or may adopt a more global view). But this depends too on the basic tools the language provides. For instance, the language may provide some capability as "put object X in queue", "extract an object from queue", etc. Otherwise the complete set of operations corresponding to queuing, etc., has to be described.

A second set of tasks to be taken into account is the whole set of operations that must be performed in the framework of any simulation study. The simplest example is "draw a random variable according to a given probability distribution". Such variables represent the duration of a task, the time interval between events such as node failures, etc. These operations (draw random

variables, manage the event list, perform basic statistics,…) are repeated for each simulation experiment.

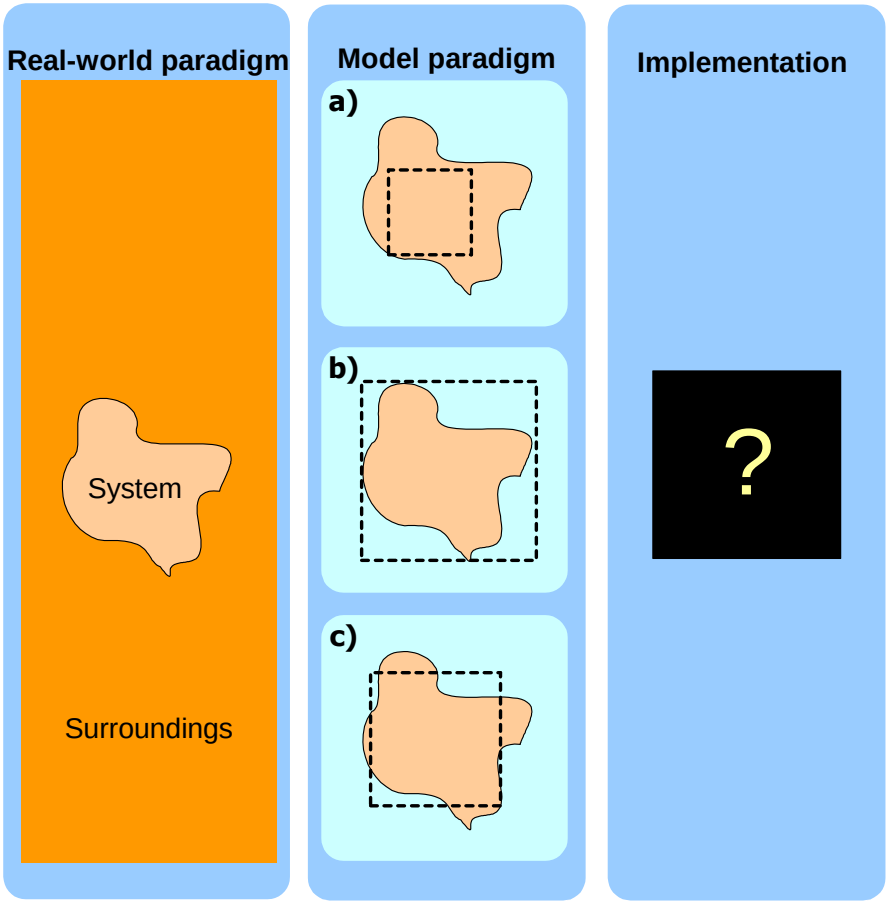


Figure 18: Modeling is an art...

Figure 18 depicts various strategies for modeling – the art is to map the real world into a simple model. Three options within the modeling paradigm are depicted:

a) Not all features are covered – simpler than the real world.

b) The model contains many features; the model has now become more complex than the real world.

c) Only a selected set of features from the real world is being modeled. In addition some extra features are being added to ease the implementation of the model.

Building the model requires in-depth knowledge of the systems to be modeled. One example of features that can be safely ignored is the data transport in networks in cases where only the control system is of interest.

Steps in modeling:

- Problem definition / setup goals. A rule of thumb for setting up goals is that the goals should be *SMART* goals, meaning that they be Specific, Measurable, Achievable, Repeatable and Thorough (all cases should be covered) [Jai91]
- Conceptual model design. Includes considerations on which features to include / exclude.
- Model translation / implementation. The actual programming task of implementing the model on a specific computer platform / tool.
- Verification & Validation. Checking that everything is correct.
- Run simulation(s)

5.1.2. Mixed complexity modeling

When doing modeling of communication systems one has the advantage that the system has already been subdivided by means of the protocol layering into useful chunks. This protocol layering should be exploited when doing modeling. Again, by careful analysis of the complete system one can determine the important parts to include in the models. Taking the list of required functionality as the outcome of the analysis it might become apparent that not all protocols in the real world are required in the model. By extracting information on which functionality is required the overall model can be greatly simplified. Hence some protocols must be modeled in full detail, some in simplified versions and some might be omitted from the model.

This is a very important observation to make. In this thesis this is called mixed complexity modeling and will be used in all simulation studies presented.

5.2. Inside simulation tools

In simulation there are two notions of time - simulated time and real time. Simulated time refers to the time frame being modeled whereas real time is the time it actually takes to complete the simulation given some implementation of the model and some computer equipment. The real time can be either longer or shorter than the simulated time. The relation solely depends on the implementation and the amount of processing power available. Gen-

erally simple models will run faster and hence it is possible to trade-off simulation duration and accuracy.

Simulators can be either time-driven or event driven. Time driven refers to simulators where at certain time intervals some processing is done. I.e., this kind of simulation is particularly well suited for simulation of systems where events must be processed at regular intervals. Examples are time division multiplex systems. Opposed to that the events driven systems will skip time-frames in which nothing happens in the system. If the systems events are sparse or just grouped this might be the simulator of choice.

5.2.1. Event driven simulation

An event refers to a change in system state. Events are processed as they are fetched from the queue one by one. Event processing involves a task to be performed and possibly generation of new events. The central entity of an event driven simulator is the event-scheduler, which is a priority queue responsible for taking care of pending events and controls the sequence of event processing. Thus the simulation execution is dictated by the events in the queue – hence the name event-driven. The simulation is terminated when a preset time is reached. Since times of inactivity are skipped this is usually referred to as compressed time.

Random numbers play a vital role in simulations. They are used to mimic the behavior of real life systems, which often depend on some user input that is not predictable.

It is well known that random number generators are not equally good. Any generator uses some algorithm to calculate the next number in a series of pseudo-random numbers. A comparison of a number of random number generators can be found in [Hol2003].

5.2.2. Simulation speed

Obviously, simulation speed is an issue. Simulations that require excessive simulation time might simply not be interesting because the results are outdated when they be available. For event driven simulations the time required to complete a simulation (real time) is proportional to the total number of events processed during the simulation run. I.e.,

Real time = c * total number of events

The constant c is dependent on the total number of simulations. Thus, it can not be necessarily be concluded that doubling the total number of events will double the simulation time. The time to process one event is not constant

because events are different. For instance inserting a packet into a queue might be a faster operation than removing one. However, *real time* as a function of total number of events is an monotonically increasing function, i.e., larger number of events implies longer simulation time.

Decreasing simulation time can be achieved in fundamentally two ways either by reducing the amount of required processing or by increasing the amount of processing power available.

Reducing the amount of required processing can be done either by reducing the number of events or by reducing the processing required for each individual event.

Parallel / distributed simulation is an obvious method for making more processing power available. Since an event driven simulation is executed by processing a number of discrete events it is possible to distribute the events among more than one processor either within the same computer (parallel execution) or more computers (distributed computing). [Szy2003][Fuji2003]

Distributing events, however, is not that simple because there is the risk of committing causality errors. A causality error occurs in the case where an event with a timestamp earlier than the current time and impacting the current process is being generated. If two or more events are scheduled for execution at the exact same time the case is straightforward; Separate processors can execute the events. All other cases require some consideration on how to avoid causality errors. Generally, distributing events among more processors requires a central scheduling unit that can handle causality. Two approaches exist: conservative and optimistic. The conservative approach avoids causality errors completely whereas the optimistic approach merely detects errors and then reverses the simulation in order to undo the event that caused the violation of causality and thus annihilate the error.

The conservative approaches works with a window within which no causality errors can occur. This implies that events scheduled for execution within this window can be safely distributed. A common way of determining the window size is by setting it to the propagation delay between two entities in the system modeled. For example in a network model the window size could be set to the propagation time between two network nodes. [Tin1989][Lub89]

5.2.3. Example tools

There are a number of tools available. By *simulation tool* is meant today not only the basic tool which runs the simulation model, but also all related utility programs attached to any simulation project, such as graphical aids and development tools – these ones becoming of growing importance. Comparing simulation tools is to some extent impossible as they have different goals,

Example tools

WIPSIM [WIPSIM], OPNET [OPNET], NS2 [NS2], Extensible and High-Fidelity TCPIP Network Simulator [Exten], MPLS Network Simulator [MPLSSim], Bluehoc [Bluehoc], CDMA Wireless Network Simulator [CDMASim], GloMoSim [GLOMOSIM], QualNet [QUALNET], CNET [CNET], Real [REAL], NetSIM [NetSIM], FLAN [FLAN], NCTUns [NCTUns], SimMan 1.0 [SimMAN], VENUS [VENUS], AnSIM [AnSIM], NIST [NIST], INSANE [INSANE].

different fields of application, offer different tools and possibly address different populations of users.

5.3. Types of simulation tools

When starting modeling one has to make a decision to either use a general-purpose language or to use a general-purpose simulation tool. The pros and cons of each approach are highlighted subsequently. The taxonomy used here is based on type of tool and whether a general or special purpose language is being used.

A first step towards the classification is in the nature of the language the simulation makes use of. To better explain it, and to understand the implication of the choice, one has to think about the tasks involved in running a simulated model of any system.

Other classifications could be proposed, e.g., emphasizing the technical aspects of the simulation kernel. For instance, some tools use parallel simulation. Other tools are presented as based on an "object oriented" approach – but this is not, however, a sound criterion, as most simulation languages inherently use these concepts.

5.3.1. Using general-purpose languages

First, the development of the simulation study may be done using general-purpose languages – such as C, C++, Fortran, etc. As they are not oriented towards simulation tasks, they offer no help for that goal: the developer has to perform the whole set of actions described above: whole description of the system in its finest details, and description of all "simulation-related" tasks. As it allows building a simulation program perfectly tailored to the needs of the study, the product obtained will have the highest possible performance level (e.g., in terms of run time). Usually, the choice of this approach is motivated by the need of extensive use of a program that would be otherwise pro-

hibitively slow. However, the gain in the exploitation phase is balanced by an increased effort in the development phase (both in terms of analyzing the model, of coding it, and last but not least, of debugging). The task of simulation development consists often in building (or assembling) a library of basic routines, with which the final package is constituted. One finds in the literature numerous examples of such libraries (see e.g., [Feldman]).

5.3.2. Using a general-purpose simulation tools

Here, the developer makes use of a simulation language, which is a computer language aiming at easing to describe the model, by providing a high-level instruction set by which the system is much more easily described than using general-purpose languages. Typical instructions allow to enter or extract "customers" from queues, to choose a service discipline, to synchronize tasks, to draw random variables, etc.

First simulation tools were proposed in the 60's, and look like general-purpose languages of the same period. Examples of such tools are Simula, GPSS and Simscript. Some of them have had a quite long career and have been continuously improved. However, most of today's users prefer tools of the following generation, characterized by a more or less sophisticated graphical interface. The simulation model may be built from the interface – through a few "mouse clicks", and the tool often provides utilities to visualize the results, and even to produce the final report. The trend is also to provide a larger and larger library containing built-in sub-models. OPNET [OPNET] is perhaps the most widely known example of this category, but many others exist (e.g., Bones, SES-Workbench).

In fact, the difference between these two categories tends to vanish. First, even if using a "genuine graphic" simulation tool, the study of any elaborated model (in fact, any model of real size, apart from toy cases) asks the developer to "open" the basic building blocks and to write down pieces of code (most frequently using C, or C++). Second, most of the languages in text form of the 60's, which are still in use, have been greatly improved and provide most of the functionality as true graphic tools. This is especially the case for Simscript II.5 (the latest version of the popular language), but other ones have evolved the same way.

5.3.3. Special-purpose simulation tools

While general-purpose simulation languages are not specific of an application, the third category offers languages through which the user simply describes the system by specifying the topology, the kind of equipment, the

numerical figures of traffics, etc. The simulator is tailored to the study of a quite specific application, such as a data network, and is of no help outside of this application. The effort of development is minimum and restricted to the definition of the simulation experiment and the analysis of the results. Examples of such tools are COMNET, SIMFACTORY (simulation of manufacturing applications), NETWORK II.5 (from CACI), etc. These tools are sometimes referred to as simulators (as opposed to simulation languages of the previous section) – see e.g., [Law][Kelton]

However, such a tool is of limited help, in that it can only be used for studying existing and well-documented technologies. It is thus poorly suited as soon as new equipment, new protocols, new networking paradigms are concerned. Rather, its field of application is to be found on pure network planning and dimensioning, in the operational phase of the technology.

The above classification appears rather "rough". The frontier between general-purpose and special-purpose simulation tools is somehow fuzzy. For instance, one may build network models using OPNET in a way much like a special-purpose language (using the specific libraries it provides), nevertheless it has to be seen as a general-purpose simulation tool, as new network devices may be freely developed.

However, the classification emphasizes a major difference between languages.

There is however a class of special-purpose tools which may be of more general help: free software tools have been devised, mostly by U.S. universities. Examples are NETSIM (MIT), NIST (for ATM networks, and based upon the previous one), INSANE (Berkeley), NS (project VINT), etc. They offer the user the possibility to develop new modules or alter the code already produced, allowing thus to enlarge the scope of the tool. NS is probably the most known example of this class.

5.3.3.1. OPNET

One example of a simulation tool is OPNET, which will be presented in some detail here because it was used for all the simulation studies presented in this thesis. OPNET (Optimum Performance Network Engineering Tool) modeler is a commercially available product from OPNET Technologies Inc. [OPNET]. OPNET is a tool equipped with a GUI and contains a (high) number of various editors for creating/modifying/verifying models and for running simulations and displaying/analyzing results. OPNET runs on top of a C compiler. Models in OPNET are built in a hierarchical fashion. Models can be built either top-down or bottom-up and each level represents the internal structure and functionality of the level above. The levels are:

- Network level (not related to OSI layer 3!)
  Modeling of network topologies and overall configuration takes place at this level of modeling. Network elements such as communication links and node devices are used to build the model. In addition, node/link failure/recovery can be modeled.

- Node level.
  At this level the internal structure of network level devices are modeled. Elements used for modeling includes: generic processor modules, queue modules, receivers and transmitters. These are interconnected by streams or statistic wires.

- Process level
  Node level device functionality is modeled at the process level. By means of finite-state-machines (FSMs) any functionality can be modeled quite efficiently.

- Proto-C level.
  The lowermost modeling level is called the proto-C level. Proto-C is an extension of the C (or C++, depending on the underlying compiler) programming language. A large number of kernel procedures are available. All built-in models are available at this level, i.e., as source code.

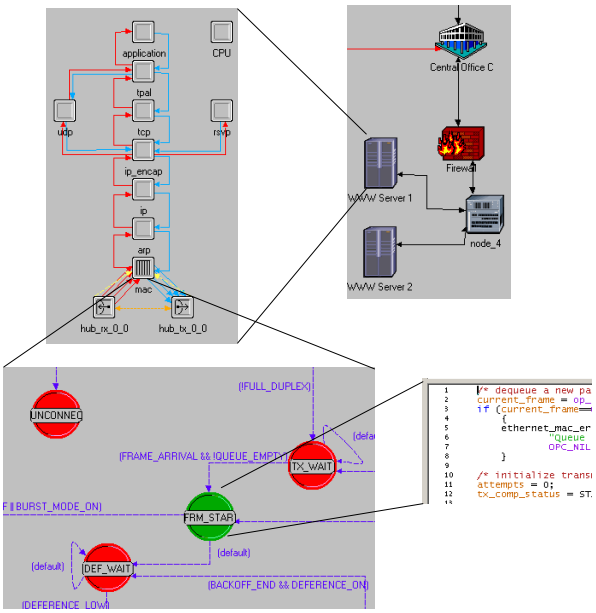


Figure 19: The hierarchical structure of OPNET models

Models can be created/edited at each level and by combining such models it is possible to define models of very complex systems. This allows modeling every single detail from layer 1 to layer 7 and on top of that, device configuration and user behavior (used for traffic modeling).

A vast number of protocol models as well as device models are available. These can be used as is or be used as a basis for model development. Everything is customizable and configurable because and OPNET is inherently extendible because it runs on top of a C-compiler. All provided models are available as source code, and can thus form a basis for further model development.

Simulation is carried out from the GUI or from the command line. Before running a simulation, the desired statistics to be collected are selected. During simulation the statistics are written to files – either scalar files or vector files depending on the type of data.

The simulation kernel itself is extremely efficient – beginning with version 10.0 the product has been optimized and runs on multi-processor computers.

OPNET has strong support for performance evaluation. It is possible to specify sets of simulation (Simulation sequences) and thus sweep parameters through a range of input parameters / traffic. The simulations can either use the built-in random number generators or one can provide / specify another. In terms of output analysis, either the built-in features for statistics can be used or all data can be exported to an external analysis tool.

5.3.4. Pros and cons

This is a summary of the overview of OPNET. All simulations in this thesis have been carried out with OPNET.

Main pros of OPNET

- Large customer base. This gives the tool quite some credibility. A high number of user means that many independent people validate the models. Because OPNET Inc. as a company has some interest in ensuring that their models are correct errors are usually corrected very fast. Hence, there is a large probability that a given model works correctly.
- Professional support.
- Very well documented.
- Ships with a large number of built-in protocols.

Main cons of OPNET

- Relatively high price – but cheap for universities…
- Complex, takes time to learn.

5.4. Network modeling

Obviously, in this thesis modeling of networks is of prime concern. However, all general modeling concepts still apply only some special considerations are required. Modeling networks is generally much harder compared to modeling, e.g., a strictly physical system, because no simple, physical laws are available and due to the scale and dynamics of large networks [Flo2001]

To model a network, methods for modeling the network topology, traffic and protocols are required. In general modeling terms traffic can be considered as the load of the system.

5.4.1. Topology modeling

There are numerous scenarios that could benefit from automated generation of topologies, including:

- routing analysis
- routing protocol evaluation
- traffic- and network- engineering
- signaling protocols
- recovery mechanisms
- introduction and implementation of innovative technologies (e.g., GMPLS or optical techniques)
- network scalability and flexibility
- network infrastructure requirements (e.g., in terms of equipment)
- traffic analysis and distribution in the network

As an example, a routing protocol may be evaluated in terms of efficiency and performance in networks with different topologies.

When modeling topologies it is common to use a graph representation of the network topology.

Below are some results based on [p10].

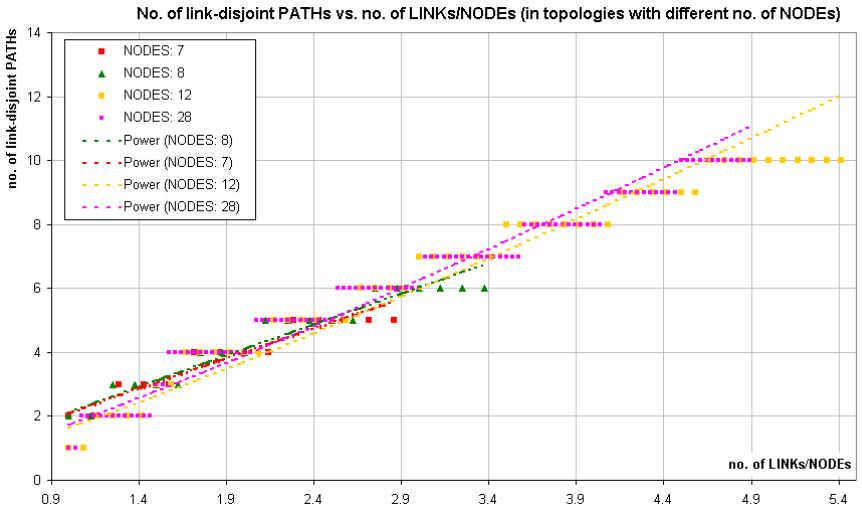


Figure 20: Link-disjoint paths scaling in terms of links/nodes

Figure 20 depicts the number of link disjoint paths versus the size of the network. Disjoint pathss are useful for protection / restoration schemes. The x-axis shows the number of links, number of nodes ratio and is hence a measure of how connected the network is. Not surprisingly, more links increases the number of disjoint paths.

Figure 21 shows the average node degree (number of links from a node) versus number of links for a number of different node counts.

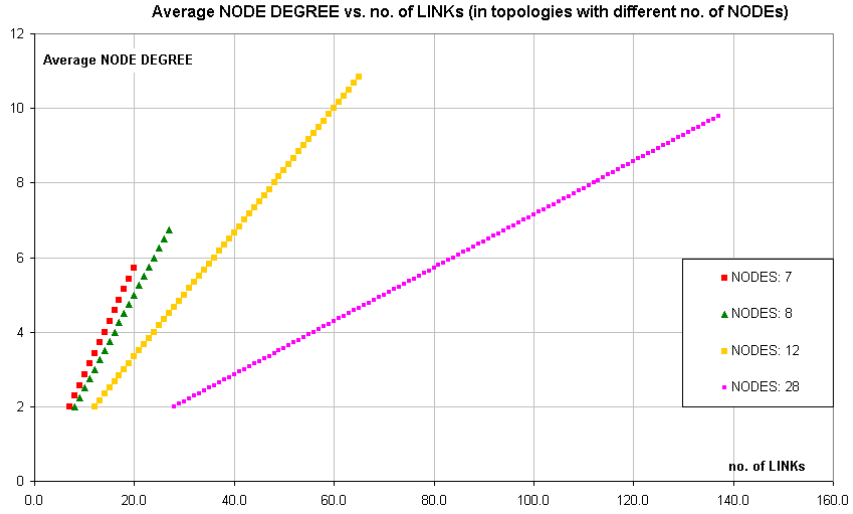


Figure 21: Average node degree in terms of links

5.4.2. Traffic modeling

Traffic can be considered at a number of levels. Raw packet streams with certain properties can be useful for e.g., queuing analysis and traffic generated by modeling real protocol behavior can be adequate for analyzing typical usage scenarios. On-off traffic source models are very common. Self-similarity (aggregated traffic) and heavy-tailed duration (user traffic) can be adequately modeled by using the heavy-tailed ON/OFF model. [Wil1997]

OPNET modeler has a number of built-in traffic generators that can generate realistic application traffic.

5.4.3. Protocol behavioral modeling

In OPNET, processes are modeled at the process level. Processes use Finite State Machines (FSMs) described by State Transition Diagrams (STDs) to define their behavior. A number of methodologies can be used to devise the FSM from a given protocol specification. The problem is to find the set of all possible states and all transitions between them – finding all the states is usually the biggest problem. A state must be mutually exclusive of and complementary to other states. Additionally, all events (which trigger changes to the system's state) must be covered by the transitions. The following algorithm can be conveniently used to determine the set of states and the transitions given only one initial state and the set of events and conditions. In Step 5 either existing states or new states are considered. New states are added until the algorithm terminates and the FSM is hence completed.

Step 1: Choose a state

Step 2: Choose an event

Step 3: Choose a condition under which event occurs

Step 4: Determine all actions to perform

Step 5: Determine final state

Loop Step 3 for all conditions

Loop Step 2 for all events

Loop Step 1 until all states are complete

5.5. Validation and verification

Correctness of the model is (of course!) of prime interest to the modeler. Correctness can be assured through series of more or less formal steps. Generally, this checking how the model's output confirms to the real world can be subdivided into two processes: validation and verification, where

Validation – building the right model

Verification – building the model right

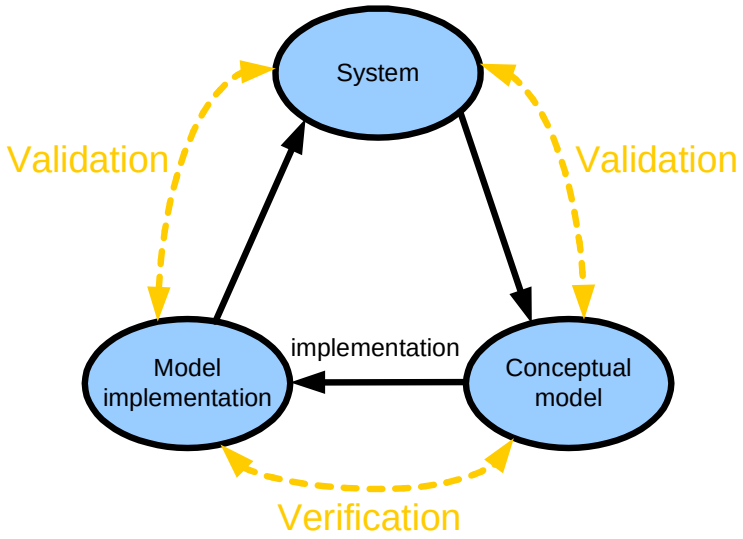


Figure 22 Model development, validation and verification (Based on [Sar2003])

Figure 22 depicts the relationship between the real-life system, the conceptual model and the implementation of the model. The verification process should ensure that the implemented model behaves as anticipated. This can be done by using ordinary software engineering methods such as debugging.

However, the implementation might not reflect the real-life just because the verification shows that the implementation is correct. If the conceptual model is not a true representation of the real world, then there will be a mismatch between the real world and the implementation.

There are a number fo techniques that can be used for validation and verification. As well static as dynamic validation techniques exist and they can be more or less formal. Among the informal methods are face validation, expert inspection and walkthroughs, which are variations on the theme of a review.

These can be complimented by e.g., sensitivity analysis, assertion checking and various statistical methods. Formal approaches are based on mathematical proof of correctness and suffer from the disadvantage that they are generally too complex to be applied to models of reasonable size. They might, however, be used to establish test methods for simpler sub-models.

It is important to note that the outcome of the Validation and verification process is not a binary variable. A model might be partially correct: Particularly when simulating networks this might be the case. [Flo2001]

However, it is a common mistake to not ensure credibility of the model [Paw2002].

5.6. Example

In order to illustrate how simplified models can still yield useful results here is one example of the modeling of a real-life satellite network. The results presented are based on the work in [p11][p12]

5.6.1. A quick overview of the system

The IPDS system was developed by Inmarsat to facilitate global access to voice and low-rate data services. The system architecture consists of land earth stations, satellites, and the users' mobile terminals, which directly communicate with the satellites, as illustrated in Figure 23

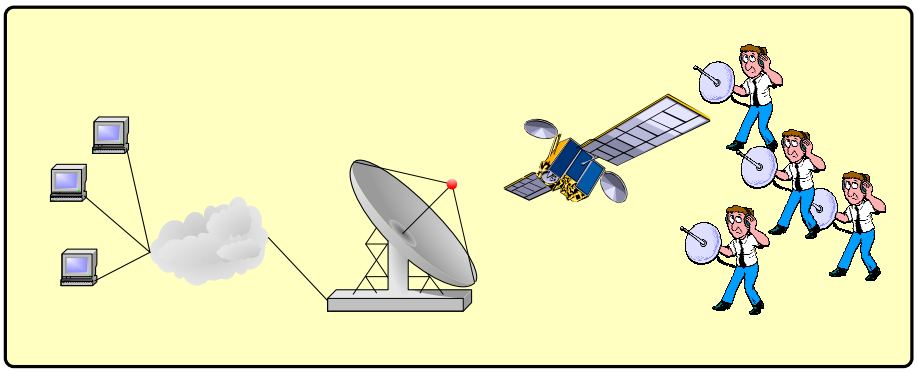


Figure 23: Example satellite network to be modeled

The functionality of the SBS (Satellite Base Station) and the MESs (Mobile Earth Stations) is divided into a number of elements for traffic handling, signaling and management. Since the scope of this modeling is to investigate the traffic performance of the IPDS system, only the network elements for traffic handling are modeled. The stationary (i.e., the SBS) and the mobile

stations are referred to as the Satellite Access Node (SAN) and Mobile Access Nodes (MANs), respectively. The basic protocol architecture for the SAN and MAN is illustrated in Figure 24.

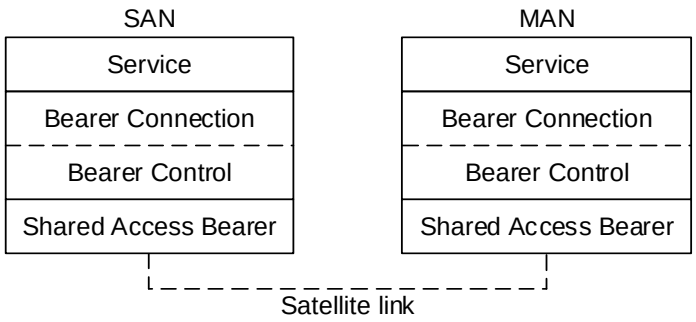


Figure 24: Satellite system protocol Architecture

The Service layer is mainly an interface to higher layers in the protocol architecture, e.g., IP or other network layer protocols in the case of data services.

The Bearer Connection sublayer is responsible for providing connection oriented logical channels across the satellite bearers, referred to as bearer connections in the rest of this paper. These bearer connections can be grouped into two categories: 1) *Automatic Repeat reQuest* (ARQ) bearer connections with an advanced selective retransmission scheme for handling transmission errors, and 2) simple bearer connections with no retransmission in case of transmission errors. In the last case, higher layers, i.e., the layers above the Service layer, must handle any retransmissions in the case of transmission errors.

The Bearer Control sublayer manages the physical (satellite) channels, which include:

- Multiplexing of the bearer connections provided by the connection sublayer onto the frames on the physical channels.
- Assignment of physical channels to the individual mobile terminals.

The Bearer Connection sublayer and the Bearer Control sublayer together are equivalent to the OSI Data Link layer.

The Shared Access Bearer is the (shared) satellite physical channel, which is shared among a group of mobile terminals. It is the task of the Bearer Control sublayer to map the bearer connections in the mobile terminals onto specific physical channels. A bearer connection's access to a satellite physical channel is controlled by the Bearer Control in the SAN based on information such as:

- The Quality-of-Service requirements of the bearer connection

- The amount of information waiting to be transmitted in a bearer connection

The Bearer Connection will keep the Bearer Control informed on the amount of information waiting to be transmitted. Based in this information, the Bearer Control at the SAN will decide which connection will be permitted to send information, and how much. The Bearer Control sublayer will send information on the physical channels in form of fixed length frames of either 20 ms or 80 ms.

5.6.2. Performance issues of TCP over satellite systems

It is a well-known issue, that the performance of TCP can be negatively affected if the transmission path has a high bandwidth-delay product, due to a limitation on the number of transmitted but unacknowledged bytes. Furthermore, the mandatory slow-start procedure used to reduce the possibility of network congestion (in terrestrial networks) also has a negative impact on the TCP performance on a high-delay path [Allman2000][Shep97]. Suggestions to improve the TCP performance on satellite links include the TCP window scale option [Jac1992], using a larger initial window [Allman2002] and using selective acknowledgments [Mathis1996]. Furthermore, the performance of TCP on satellite links has an implication on the applications that uses TCP, e.g., FTP, HTTP and email. See for instance [Krus2001] for the performance of HTTP over satellite channels.

The following main features characterize a satellite system:

- Large delay (due to the geostationary position of the satellites)
- Mobile nodes that communicate directly with the satellites
- Stationary Earth Stations

The system studied here consists of geostationary satellites, Satellite earth stations, and mobile terminals.

So this is the real life system, which is obviously very complex. The trick is now to find an approach for doing a simple model.

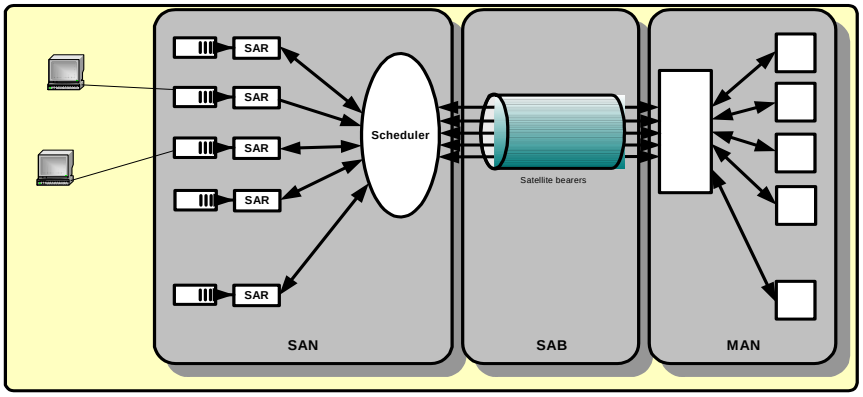


Figure 25: The abstract model of the satellite system

The model has been simplified in a number of areas. It models the data transport only and thus ignores the control part of the satellite system. However, the model has also been extended with features, which are not present in the real life system. At the client side a hub device has been added to distribute packets among the portable satellite receivers. The real life system (of course) does not have such a device; it uses an advanced frequency allocation procedure to ensure that the devices use different frequencies. In the model, however, this control feature is not being modeled (since it only impact the start-up procedure of the device and thus have no impact on the data transport). Moreover it doesn't degrade the simulation performance.

The modeling was based on the *mixed complexity modeling* approach. Only part of the lower (1+2) layers was modeled, but higher layers were modeled in full detail. The development benefited from a number of built-in models in OPNET for instance the TCP model and the application models.

5.6.3. Results

In this section sample results from the satellite study are shown. The results are an example of protocol modeling. The model was implemented in OPNET modeler version 9.0. The application tested here is a web-browsing application with browser and server running HTTP 1.1. Each web page requested consists of a 500 bytes HTML page plus 5 small pictures of sizes between 50 and 400 bytes. More results can be found in [p11][p12].

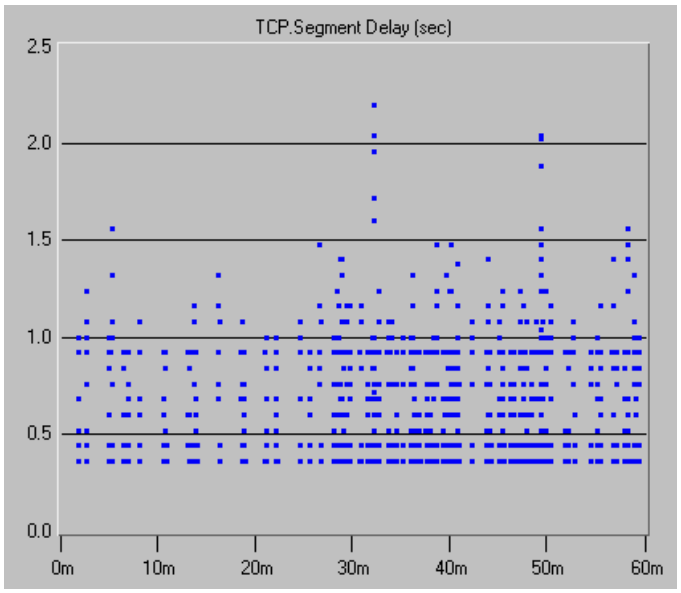


Figure 26: Individual TCP segment delays

Figure 26 and Figure 27 show examples of results from this modelling work. Both graphs show results for 1 hour of simulation (hence the scale 60 minutes on the x axis).

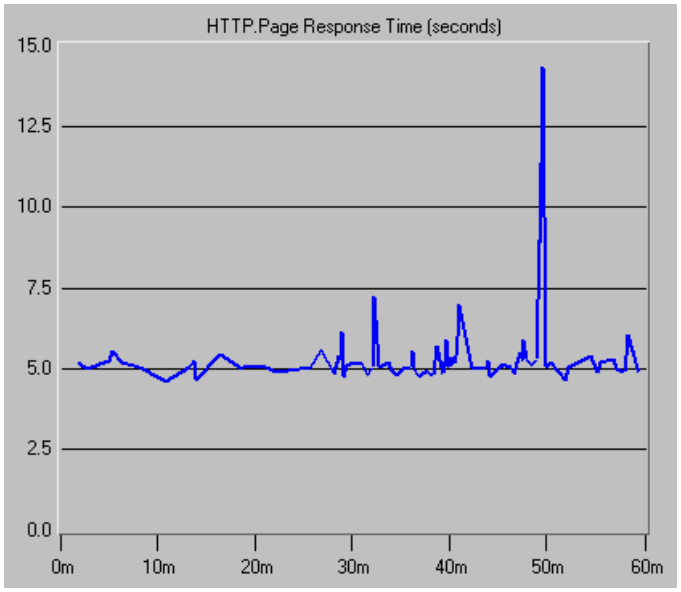


Figure 27: HTTP page response time

5.7. Pros and cons of modeling

Modeling is not always the best choice for performance evaluation. Below is a list of reasons for *not* using simulation: [Banks97]

- Using common sense can solve the problem.
- The problem can be solved analytically.
- It's easier to perform measurements on the real system.
- The cost of the simulation exceeds possible savings.
- Insufficient resources are available for the project.
- Insufficient time for the simulation project.
- No data – not even estimates – available.
- The model can't be verified or validated.
- Project expectations can't be met.
- The system under test is too complex or can't be modeled.

5.8. Alternative approaches

Although simulation is a very useful technique alternatives do exist.

- Emulation
- Analytical solution / calculations
- Measurements

5.9. Summary

This chapter has presented an overview of simulation, its pros and cons as well as some strategies for building models. The idea of mixed complexity modeling was introduced. Exploiting the layered architecture of communication systems can be a shortcut to model simplification. Mixed complexity modeling can be used to simplify network models. Finally, an example based on the study of a real life satellite network was presented.

6. Modeling node behavior in a network context

"The cure for boredom is curiosity. There is no cure for curiosity."

D. Parker

Nodes are *interconnection devices* and as such play a major role in ensuring *flexibility* in networks. When interconnecting two domains at least one node must work as a flexibility point – an adaptation device, which can effectively decouple the domains.

Modeling network devices and their behavior in a real network can be very complex. First of all the devices are complex and their behavior and its impact on real network traffic is very complex. Complex devices or traffic characteristics are hard to model, mainly because the models are hard to verify. Second, al full-blown analysis is often impractical due to the sheer amount of data generated from such an analysis. Mixed complexity modeling can be employed to simplify the approach. This might give useful results, but extrapolating to full network view is still virtually impossible.

This chapter covers modeling of two functions in network nodes: traffic aggregation and support for packet forwarding without any modification of the packets. These concepts are particularly useful in hierarchical architectures such as mixed electrical/optical architectures, but also applications to e.g., mobile networks are presented.

6.1. Aggregation / adaptation devices

Interconnecting network domains requires flexible gateways or adaptations units that can effectively decouple differences among the domains and ensure efficient data transport and interoperability. Of primary concern here are network domains employing electrical and optical technologies, respectively. When the technology sets the fundamental limit for switching speed in terms

of packets per second some considerations must be made on how to interconnect. The minimum packet length depends on the bit rate and the maximum number of packet that can be handled per second. Thus, when using optical technologies where very high bit rates are possible packets must be rather large in order not to waste capacity. This section considers the problem of constructing such large packets from other smaller packets and how that impacts application behavior. The results are mainly based on the work done in [p04][p09].

6.1.1. Real world aggregation devices

The traffic aggregation unit is shown in detail in Figure 28. As can be seen it interconnects two network domains and is thus able to adapt the traffic flowing from one domain to the other. As shown in the figure there are M traffic streams going into the aggregation unit. The unit contains M traffic classifiers, a number of queues (generally one per destination per QoS class) and a scheduling unit responsible for packet extraction. The figure shows one outgoing traffic stream only. In reality, however, several are needed. They have been omitted for the sake of easing figure readability.

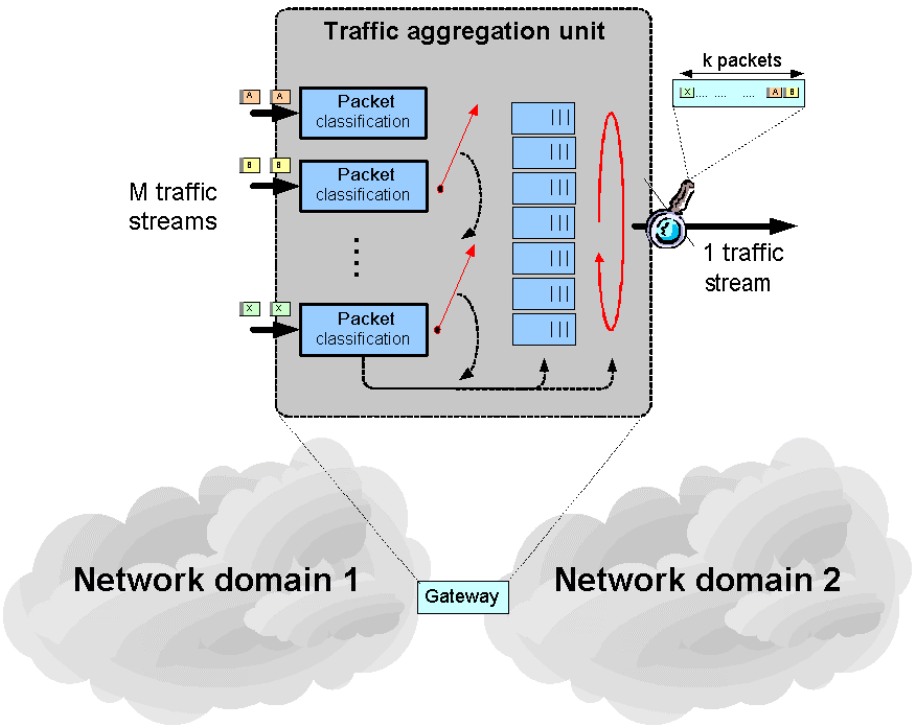


Figure 28: Details of the traffic aggregation unit

Operation of the aggregation unit is split into time units. Each time unit the output scheduler selects a queue and picks up to k packets from that queue. (i.e., if the number of packets contained in the queue is greater than k only k packets are extracted, otherwise all packets can be removed from the queue). A scheduling algorithm performs selection of the next queue.

The number of packets extracted each time unit must be larger than the number of incoming packets in order to prevent the queues from filling up and packets being dropped. Thus k > M, typically one would use k = M+1 in order not to waste too many resources on the outgoing link.

The simplest scheduling scheme possible is a round robin scheme, in which backlogged queues are selected in turn. Backlogged queues are queues containing packets, i.e., empty queues are skipped in the selection process (this makes a *work conserving* scheme, as in each timeslot at least one packet is always extracted unless all queues are empty). There is, however, a problem with this scheme, namely that the delay is unbounded. It is usually desirable that the worst-case delay be bounded so that end-to-end delay guarantees can be given. The literature is packet with examples of scheduling algorithms (e.g., [Rex96][Tho97]).

Now, consider a more elaborate queuing system (here named *'Time stamping'*), which works as follows

Packet insertion:

- Each arriving packet is time stamped
- A classification unit determines in which queue to put the packet and inserts it into that queue as it arrive.

Packet extraction:

- The packet extraction is performed once per timeslot, i.e., each timeslot the scheduling unit will output one packet (the only exception being the case in which all queues are empty).
- The output scheduler then in each timeslot selects the queue in which the head of line packet has the lowest timestamp. Up to k (incoming) packets are then extracted from that queue and bundled together to form one outgoing packet.

It has been shown analytically [Ber2002] that when using this scheduling algorithm the worst-case delay, D, is bounded by:

$$D = \left\lceil \frac{(Q-1)\cdot(k-1)}{k-M} \right\rceil$$

,where

Q is the number of queues

M is the number of input streams

k is the speed-up (maximum allowable number of packets extracted per timeslot)

But the typical delay is much smaller and depends on the input traffic characteristics. This subsequent section uses simulation to explore the behavior of such devices.

6.2. Modeling of adaptation devices

A real device would contain M packet classifiers to determine the destination queue for each incoming packet. The packet classification scheme is usually relatively complex and includes examining part of or the entire header of the

incoming packets. This requires high processing power in the forwarding engines of the switches. In real devices this is a major bottleneck.

In the simulation this is modeled simply as a field in the packets containing the queue number.

6.2.1. Modeling methodology

Simulation goals

The goals of the simulation are to examine various aggregation schemes with respect to

- Mean delay
- Delay variations
- Bundling efficiency

The maximum delay has been determined analytically and is clearly important. However, the mean delays as well as the delay distribution are important factors also.

The bundling efficiency is defined as the ratio:

$$\text{Bundling efficiency} = \frac{\text{Number of packets extracted}}{\text{Speed up}}$$

Thus, the bundling efficiency describes whether the outgoing packets are filled entirely or only partially and thus is a measure of bandwidth utilization.

6.2.2. OPNET model

In the simulations the following assumptions are made:

- The packet duration (in seconds) is the same for incoming and outgoing packets. This implies that that the speed-up directly translates into a bit rate difference. I.e. the speed-up means a difference in the number of bits contained in the incoming and outgoing packets, respectively. Such a difference can occur either by varying the packet's duration or the bit rate. For real life technologies the difference is typically 4 for telecom standards (e.g., SDH [G.803] and ATM [I.361]) and 10 for data communications standards (e.g., Ethernet)

In the simulations all delays are measured in *timeslots*, where the output scheduler bundles and sends one packet from the queues each timeslot. Thus, all delays are expressed relative to one timeslot.

The length of such a timeslot in real systems clearly depends on the technology used. Typically, the classification/scheduling processes themselves do not limit the speed but rather by the switching technology. For instance the new optical technologies such as MEMS are currently limited by technological / manufacturing constraints.

The OPNET model consists of three parts: a traffic generator, the actual scheduling unit, and a traffic analyzer. Figure 29 shows the overall structure of the model. The details are described below.

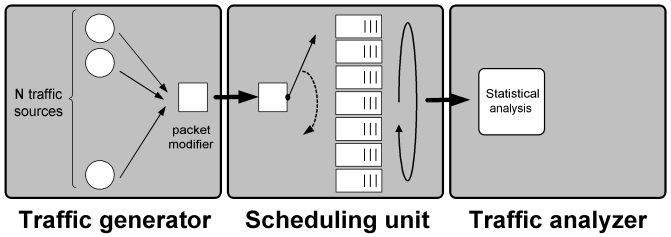


Figure 29: Overview of the OPNET model

Traffic generator

The traffic generation module uses two stages: The actual generation module, which spawns child processes for traffic generation such that traffic patterns, which are best described as sum of N sources (e.g., self-similar traffic) can be easily generated. The module generates packets with only one field, which is the destination queue in the scheduling unit. The traffic generation unit is followed by a packet modifier responsible for setting the value in the destination queue field in the packets. The values are assigned randomly based on a predefined distribution such that traffic can be either evenly distributed among the queues or some queues can be loaded more heavily than others.

Aggregation unit

The main part of the aggregation unit is a system of queues, followed by a scheduling device, which is responsible for removing packets from the queues and bundling them into larger packets, i.e., it's in this device the actual aggregation takes place. A queue extraction event forces the process to search among all queues for the head of line packet having lowest timestamp.

Traffic analyzer

The traffic analyzer just reads the incoming packets and records a number of statistics.

Model parameters

Attribute	Description
Generator	
- Number of queues	Number of queues in the scheduling unit.
- Queue distribution	How should packets be distributed among the queues?
- Source inter arrival PDF	The traffic characteristics of each source.
- Number of sources (M)	The output traffic is an aggregation of a number of sources each with the above specified traffic characteristic.
Scheduler	
- Scheduling algorithm	Choice of algorithm
- Service rate	Reference timeslot length. Equal to one in all simulations presented here.
- Speed up (k)	Max number of packets that may be extracted per timeslot. Alternatively, the speed-up can be automatically set to M+1

6.2.3. Verification and validation

Figure 30 below depicts the simulation results from a set of simulations in which the average packet delay was measured as a function of number of queues using the *'Time stamping'* scheme. Not surprisingly the delay increases with the number of queues, since the available output bandwidth must now be shared among more queues.

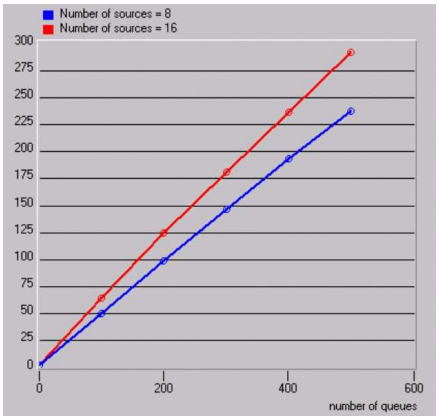


Figure 30: Mean packet delay as a function of number of queues for M=16, k=17

In addition, results from a similar analysis were published by my colleague Michael Berger in [Ber2002]. The results are identical and based on that it can safely be concluded that the model works correctly.

6.2.4. Simulation Results

A number of experiments have been carried out with the model described above. Two different scheduling schemes (round robin and 'Time stamping') have been compared under various traffic conditions.

For a fixed number of input sources it is interesting to see what impact the speed-up (k) has on the average delay. Figure 31 shows mean delay versus speed up for three different numbers of queues. The results show that the average delay decreases with speed-up initially and then stays constant when the speed-up gets greater than the number of input sources. Again, this result is intuitively correct.

All the above simulation results have been obtained for incoming packets being evenly distributed among all queues, i.e., when a packet enters the aggregation unit, the queue it enters is selected randomly (with a uniform distribution)

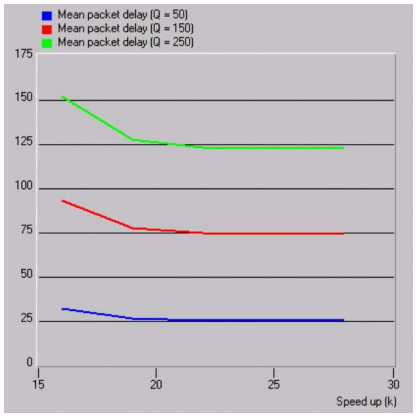
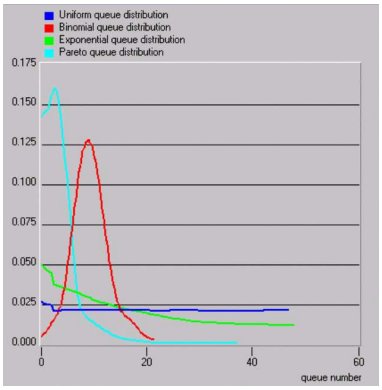


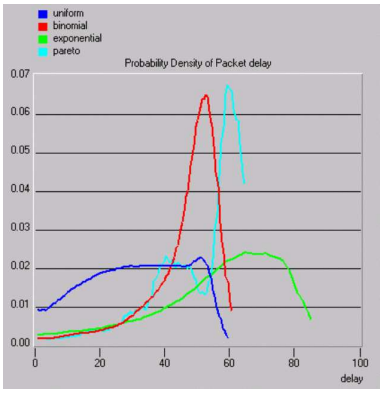
Figure 31: Mean packet delay as a function of speed up (k) for fixed M=15

Now, what if the distribution of packets among the queues is no longer uniform? The packet sources are still constant sources, i.e., they generate one packet per timeslot, but this constant flow of packets is now no longer spread out equally among the queues. The term queue distribution is used below to describe the way incoming packets are put into the system of queues. Clearly, uniform distribution means each queue getting its equal share of the incoming packets and thus corresponds to the cases presented above. Below is shown the result of simulations where four different queue distributions are used. These are depicted in the figure below (Figure 32), Figure 32 a) shows the queue distribution while Figure 32 b) shows the probability density of packet delay through the queuing system. The figure shows that the queue distribution has a huge impact on the packet delay. This is rather interesting, since the uniform distribution is not very likely in real networks. In real networks one would usually assign one queue per destination or group of destinations. If the networks supports service / traffic differentiation then it is common to use separate queues for each priority class. Hence, the number of queues must be multiplied by the number of service classes available in the network. Thus, it is very unlikely that the traffic will be equally distributed among all destinations (and also among traffic classes if QoS is used)

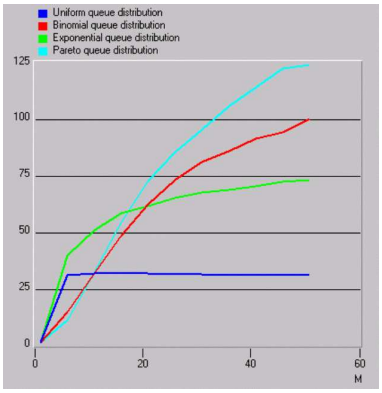
Figure 32 c) depicts the mean packet delay versus number of sources (M) for a system of 50 queues. In each simulation the speed-up was set to k=M+1. Obviously, for uniform input queue distribution the delay stays constant, but for other distributions the mean delay grows drastically, again stressing the impact the queue distribution has on mean delays.



a) Queue distribution



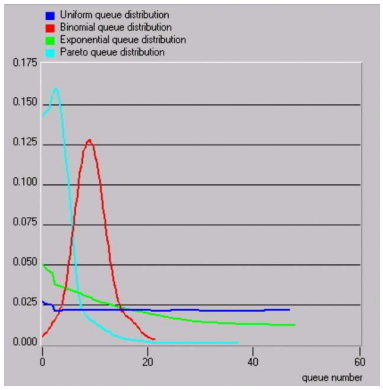
b) Delay distribution



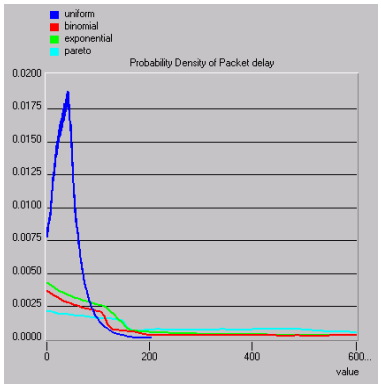
c) Mean delay versus M

Figure 32: For the Time stamping' scheme the queue distribution (a) impacts the output delay distribution (b) as well as the mean delay as a function of M with k = M+1 (c)

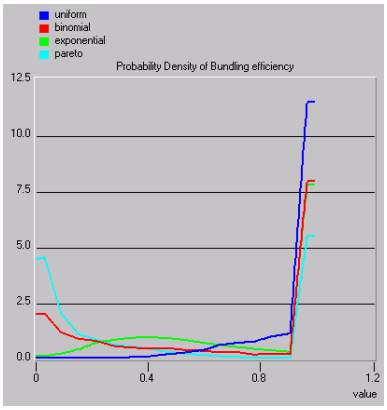
Interestingly enough when considering the round robin scheme, the results are entirely different (see Figure 33). Generally, the delays are larger but also the distribution is more flat giving a large delay variation. Figure 33 b) showing the delay distribution has been cut off to more clearly show the details for low delays but the tails of the distributions extend to several thousands!



a) Queue distribution



b) Delay distribution



c) Bundling efficiency distribution

Figure 33: In the round robin case, the queue distribution (a) impacts the output delay distribution (b) as well as the bundling efficiency distribution (c)

The bundling efficiency is shown for the round robin scheme in Figure 33 c), which shows the bundling efficiency probability density function. It clearly shows another bad property (besides the unbounded delay) of the round robin scheme. For input distributions where the packets are distributed unevenly among the queues the bundling efficiency is very poor, i.e., the bandwidth on the outgoing link is poorly utilized. In comparison, the time stamping' scheme (see Figure 34) is much better. It can be seen that the uniform queue distribution yields worst efficiency and the bundling efficiency improves with other distributions. Bundling efficiencies below 0.5 are very rare.

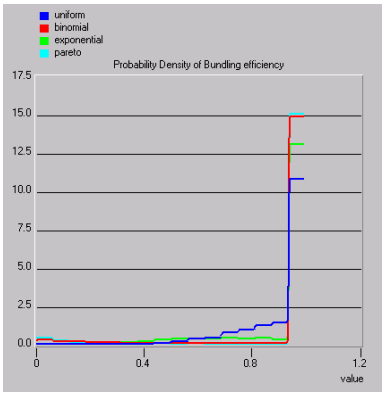
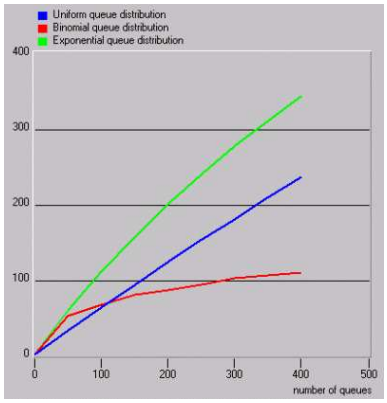
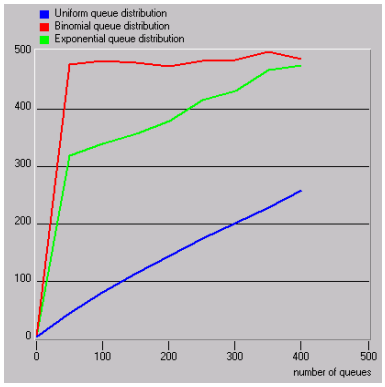


Figure 34: Bundling efficiency probability density function for the 'time stamping' scheme.

Clearly, the number of queues impacts the delay. This is shown in Figure 35 for the 'time stamping' and round robin scheduling schemes, respectively.



a) 'time stamping' scheduling



b) round robin scheduling

Figure 35: a, b: Mean delay versus number of queues for 'time stamping' and round robin scheduling schemes.

An important thing to note is that the simulated average delay (for the 'Time Stamping' scheme) is much smaller than the maximum delay derived analytically. The main reason for this being that the traffic pattern used for deriving the analytical result (the worst case scenario) is very unlikely in real networks.

Another issue is what speed-up to choose. Obviously, increasing the speed-up decreases the delay at the expense of poorer bandwidth utilization. Figure 36 shows the mean delay (for M=15 and 50 queues) versus the speed-up (k). Only the *'Time Stamping'* scheme is considered here. It can be seen that the benefit of increasing the speed-up increases for unevenly distributed packets.

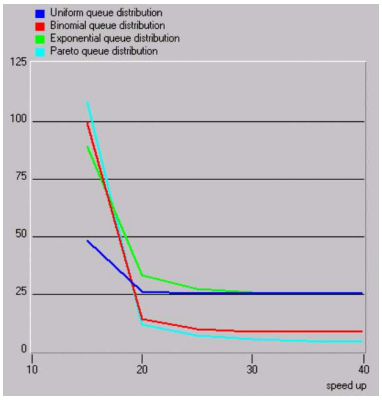


Figure 36: Mean delay versus speed up

6.2.5. TCP and application performance

The results reported previously relate to a very academic approach to queuing systems. From these results some conclusions can be made but predicting how real applications perform is almost impossible based on the results from such simulations.

A multitude of real-life applications use protocols that run on top of TCP (e.g., HTTP, SMTP). TCP has built-in congestion control mechanisms that make it adapt to the bottleneck bandwidth between server and client. When aggregating packets in a aggregation device this impacts the delay of the individual packets and hence might impact the TCP behavior. In this section mixed complexity modeling is being used to study how an aggregation device in the TCP control loop impact TCP and the applications running on top. Due to the end-to-end scheme behind the TCP design such a situation is very likely. In heterogeneous networks adaptation devices will always be present.

The model used previously is now being deployed into a more realistic environment.

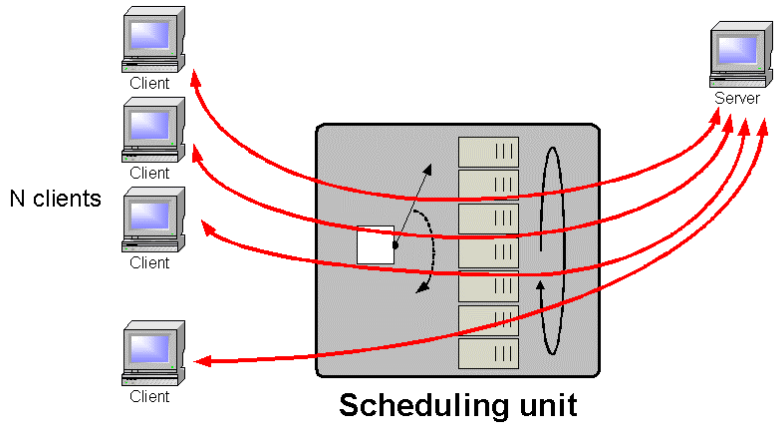


Figure 37: TCP connections on top of an aggregation unit

Figure 37 depicts the model setup. A number of TCP sessions are sharing the same bottleneck link. This is modelled in the way that they all connect to the same server. In the path between the clients and the server the aggregation device is placed.

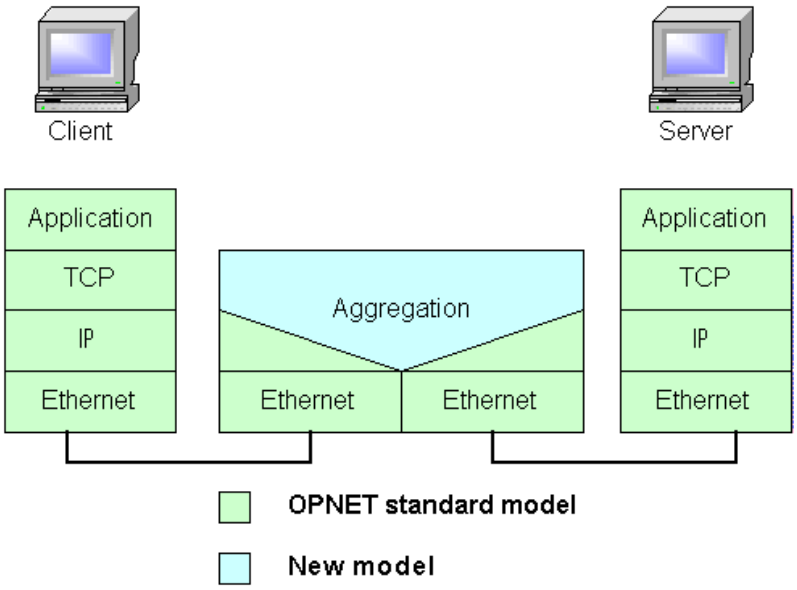


Figure 38: Implementation details of the model. Only the blue parts of the figure had to be developed, the rest were already available in OPNET

The model was implemented in OPNET version 10.0.

The results are shown below: Figure 39 depicts the probability density function (PDF) for FTP for the time stamping and the round robin, respectively. The differences are really minor, however, the tail of the round robin is significantly longer illustrating that the worst-case delay is improved by using the time stamping scheme. These results were obtained for packets of size 50k bytes. Figure 40 shows the results for the same study but using 1M bytes packets. The results are similar.

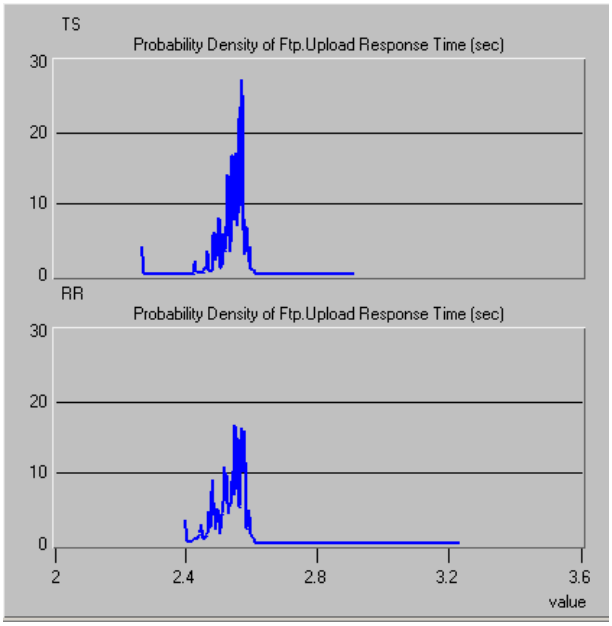


Figure 39: FTP response time PDF for 50k packets

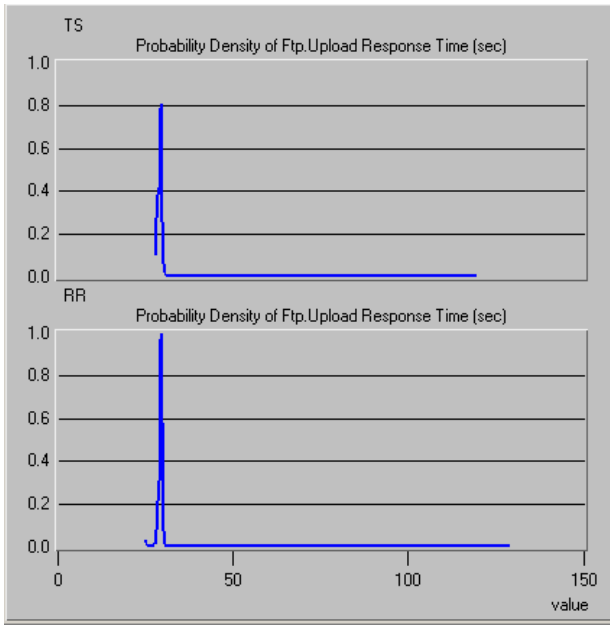


Figure 40: FTP response time PDF for 1000k packets

For HTTP the results are as depicted in Figure 41 and Figure 42. The pages downloaded in the study are of size 500 bytes HTML page plus 5 small pictures of sizes between 50 and 400 bytes.

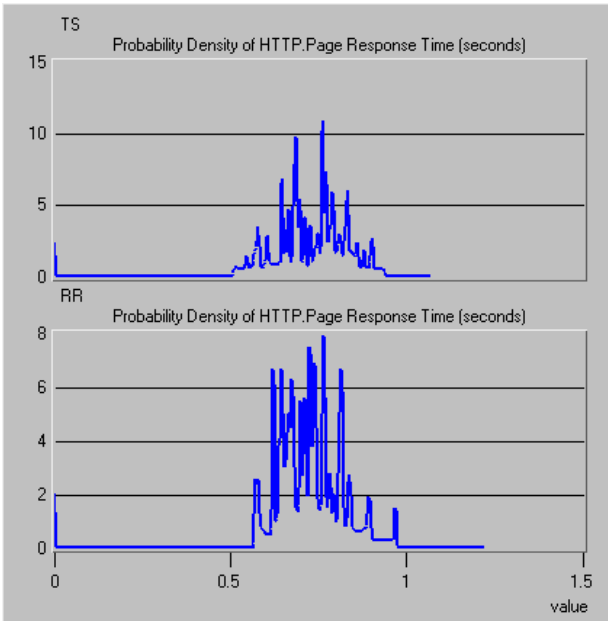


Figure 41: HTTP response time PDF for 50k packets

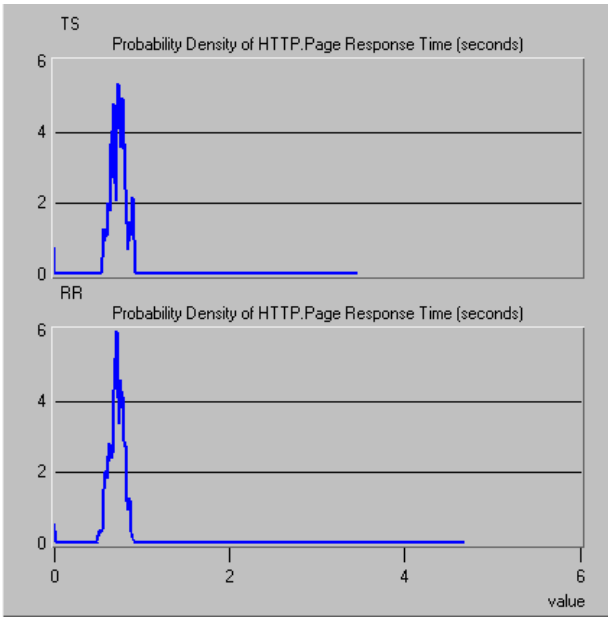


Figure 42: HTTP response time PDF for 1000k packets

The overall conclusion from this TCP study is that the impact on TCP/application behavior from the scheduling schemes investigated here are very minor.

6.3. Applications

The results presented so far seem to be of rather limited applicability. However, this is not the case! Actually, the model can be used in all situations in which smaller packets are aggregated, i.e., in all cases where network domains using different packet sizes are involved. This includes all networks where parts of the network support either larger or smaller packets than the rest pf the network. To illustrate that the same model can be used to model a number of possible applications, a number of applications are presented here. In addition, a number of application are presented in [Ber2002]

- Hierarchical MPLS, GMPLS, OBS
- ATM inverse multiplexing
- GPRS packet access
- GSM HSCSD

6.3.1. Hierarchical MPLS

Hierarchical MPLS (H-MPLS) was introduced in chapter 4. Since H-MPLS copes with heterogeneous networks that contain e.g., as well optical as electrical packet switching equipment. As discussed previously optical networks (whether OPS or OBS based) use much larger packets than their electronic counterparts. A similar study can be seen in [Gow2003][Detti2002].

6.3.2. ATM inverse multiplexing

Inverse multiplexing for ATM [IMA97] is a way of bundling together lower capacity ATM links to one virtual link with a higher capacity. Hence, at one end of the bundle of links ATM cells must be collected. Since ATM works with fixed sized packets this case is not directly equivalent to the one above but at some point within the network the cells must be bundled into larger packets in order to be useful to applications.

6.3.3. GPRS packet access

GPRS overlay was designed to allow the transport of packets in the GSM network. When transmitting a packet between the mobile node and the wired infrastructure more than one channel (timeslot) can be utilized. At the receiving end these individual bursts must be assembled to larger packets.

6.3.4. GSM HSCSD

GSM networks are characterized by a rather low capacity on the links. The reason for the low capacity is that GSM was only intended to be used for voice. However, in High Speed Circuit Switched Data (HSCSD) a number of these channels are combined to yield higher bit rates.

6.4. Key routing

In addition to queuing in network nodes modifying the packet header is one of the fundamental functionalities required. Since each packet must be processed in the network nodes, when going towards higher number of packets per second packet processing will inevitably become a bottleneck. In this section alternative approaches are investigated – approaches that try to entirely avoid packet alterations. Another possible solution was presented in the preceding section, namely to aggregate a number of packets and thus decrease the number of packets per second. The scheme presented here is named *key routing* because it is a method for finding the next hop in a sequence of packet switches based on a key contained in the packet and thus eliminating the need for a routing protocol. A prerequisite for utilizing this scheme is that connections oriented networks be used or that source routing be used. Example technologies thus are ATM, or MPLS. Constraint based routing could additionally be employed to further refine network resource utilization. This section presents a novel scheme for doing packet forwarding without packet modifications and is mainly based on [p02][p03][p15][p16]

6.4.1. Avoiding label swapping through keyword recognition

In this scheme a so-called routing-key, which is contained within each packet and which specifies the route of a packet through the network, is introduced. It requires no a priori knowledge of the route in the switches and is not changed en route, which means that no lookup tables in the switches are needed, while a label distribution protocol can be avoided. The scheme relies on a mathematical function, F, within each switch; typically identical func-

tions will be used in all switches in the network. Furthermore, an ID number identifies each switch in the network uniquely. Each switch has a function, F, which it uses to tie the ID number of the switch to the routing key in order to get the number of the output port. Using a simple example in Figure 43 to demonstrate the scheme, the following relation must in this case be satisfied for the routing key:

F(key, ID1) = P3

F(key, ID5) = P1

F(key, ID2) = P4

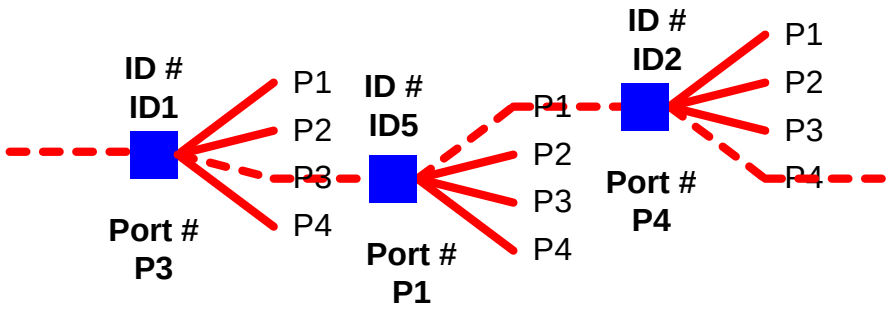


Figure 43: In general, a route through a network can be described as (ID#, port#) tuples. In the example depicted here the dashed route can be described as [(ID1,P3),(ID5,P1),(ID2,P4)]

This means that the same function and the same routing key, but different switch IDs give different port numbers. Naturally, the port numbers can be chosen arbitrarily as they depend on the chosen route for a packet, i.e., choice of route is related with choice of the key. Furthermore, different routes might be chosen for packets traveling to the same destination depending on QoS requirements. This enables traffic engineering, which is a prerequisite for providing QoS guarantees if the data traffic has long-range dependent characteristics. [Err1996]

In general, given a route described by a sequence of (switch ID, port #) tuples, the function F must meet the following criteria:

F(key, IDn) = port# n,

where IDn, is an ID number uniquely identifying switch number n and port# n is the number of the outgoing port on switch number n. This must

be satisfied for an arbitrary number of switches and one should be aware that the number of ports on different switches might be different.

If only a subset of the nodes in the network supports this key routing method then they can be grouped to form a separate MPLS domain (see Figure 44). This way of grouping nodes to form separate MPLS domains is also an integrated part of the MPLS concept [Ros1999]. The figure shows a collapsed version of a hierarchical network.

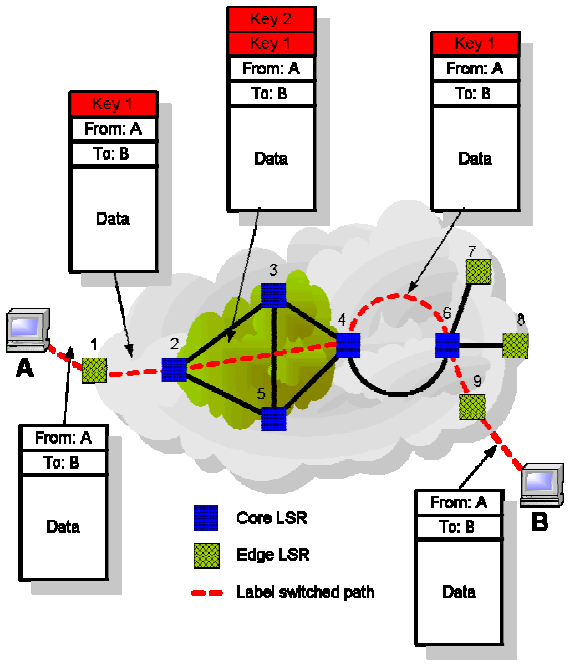


Figure 44: Multiple domains can easily be handled. When entering a new domain (as depicted by the green cloud) a new label is pushed onto the stack. The label is again popped whenever the packet leaves the domain. In this way conceptually different routing schemes can be used in each domain.

6.4.2. A key routing example

One example of a function satisfying the requirements set up above is F(key, ID) = key *mod* ID, where *mod* denotes the modulus, i.e., the remainder after integer division of key with ID. If this function is applied, a mathematical theorem exists that can easily provide the correct key values. This theorem is known by the name *Chinese Remainder Theorem* [Riv1990]. The only requirements for using this theorem is that the ID numbers, in addition to be-

ing unique, have 1 as their greatest common divisor, i.e., the ID numbers must be relative prime.

Consider the following example (see Figure 45): Here a route is passing through switches with ID numbers 5, 17 and 9 respectively. Within these switches the packets are to be routed to port 2, 12 and 5. Thus, it is necessary to find a key so that

F(key, 5) = 2, F(key, 17) = 12 and F(key, 9) = 5

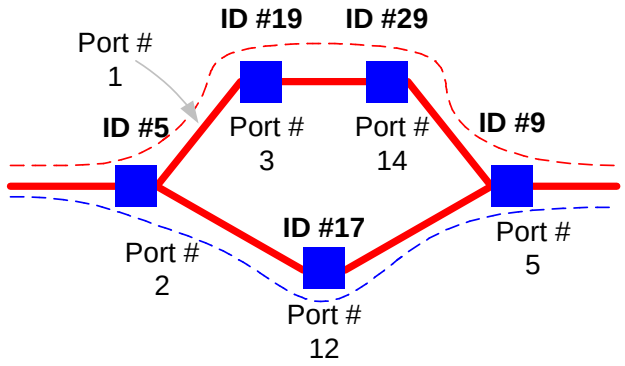


Figure 45: A subset of a network with two routes through it. The routing-key method enables selection of which route to follow and thus makes traffic engineering possible.

By employing the Chinese remainder theorem one finds that the key in this case is 437. It is easily verified that

437 mod 5 = 2, 437 mod 17 = 12 and 437 mod 9 = 5

Thus, all switches in this example will be able to route the packets containing this key correctly. Suppose that based on traffic measurements it is decided to also use an alternate route to the same destination so that the load can be balanced. Selecting the alternative route though the switches 19 and 29 only require a different routing key, which by Chinese Remainder Theorem is 21416. Testing this key against switch IDs 5, 19, 29 and 9 one gets:

21416 mod 5 = 1,

21416 mod 19 = 3,

21416 mod 29 = 14 and

21416 mod 9 = 5,

i.e., by selecting the right routing key, one can pinpoint the route of a packet through a network without having to invoke a signaling protocol to set up the switches.

6.5. Modeling of key routing

The key routing scheme, using the Chinese remainder theorem, together with a method for automatically generating topologies was implemented in OPNET. This section describes the modeling and the simulation results and is based primarily on the work done in [p08]

The real-life network must be simplified greatly in order to be able to build a model that can produce results within an accept-able timeframe. A brute-force modeling methodology that just tries to model the real network in every detail is inappropriate. Below the goals for the simulation are identified and based on that the simplified simulation model can be set up. Obviously, the model must be simple enough to achieve the identified goals, while representing a fair model of the real network.

The goal of this simulation study is to build a model of how GMPLS interacts with an MPLS based network. With the model it should be possible to measure/study:

- Call setup probability
- Network topology / routing issues
- Label length required for key routing

A list of input parameters is provided below:

<table>
<tr><th>Attribute</th><th>Description</th></tr>
<tr><td colspan="2">Topology generation parameters</td></tr>
<tr><td>- Number of nodes</td><td rowspan="3">Size and connectivity of the network</td></tr>
<tr><td>- Number of links</td></tr>
<tr><td>- Maximum distance</td></tr>
<tr><td>Path constraints</td><td>Bandwidth constraints</td></tr>
<tr><td>Type of network</td><td>SONET / pure optical</td></tr>
</table>

OPNET implementation

The GMPLS implementation has been made with OPNET modeler 8.0 and the MPLS model suite. The MPLS model has been extended/modified in order to create a GMPLS network element that can be built into MPLS network. This GMPLS models element represents the entire GMPLS network, i.e., a complete topology can be built with this single node. Figure 46 illustrates how the GMPLS network can interoperate with MPLS devices, i.e., LSPs can be setup through the GMPLS domain in this mixed environment.

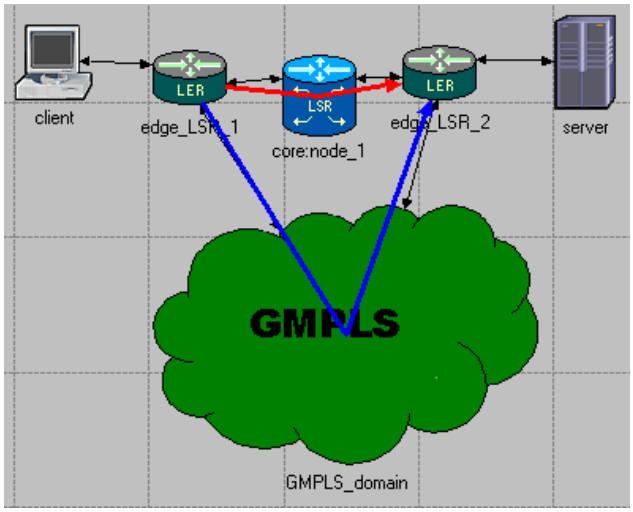


Figure 46: The GMPLS models fits nicely into the MPLS network models provided by OPNET. Hence, heterogeneous networks can be simulated.

A number of modifications to the OPNET MPLS models are needed. As well the user as the control plane need to be modified.

In order to minimize the modifications needed in the OPNET code, GMPLS has been implemented as a separate process within the network nodes. The LDP process has then just been modified to detect whether the GMPLS process is present or not (and hence whether this is a MPLS or GMPLS node)

The GMPLS model mimics the entire GMPLS network domain. I.e., even though it only appears as one single node in the figure, to the implementation it functions as a complete sub-network. The size of the GMPLS domain is user configurable.

Topology generation is performed by using the *Route package* in OPNET. By using an implementation of the Dijkstra SPF algorithm [Riv1990] full connectivity of the generated topology is ensured. The GMPLS implementations allows for either topology import from a file or generation of arbitrary topologies based on a specification of the networks size (number of nodes and links). Modeling network topologies has been studied by a number of researchers [Zeg1996][Fen2000] and it has been shown that the topologies

have an impact on the network behavior. The topologies generated are well suited to model an optical WDM network, i.e., the capacity of each link is given as a number of wavelengths. The actual capacity (i.e., bit rate) of each wavelength is not modeled explicitly. This is not necessary when path setup only is considered as in this study, the important thing here is whether a path (e.g., wavelength) is available or not.

The setup state tries to find a route through the network. One path requires one available wavelength from source to destination node. An attempt is made to find the shortest possible path though the network. This minimizes the overall capacity consumption of the path and moreover (if the network is build from optical cross-connects where requirements for e.g., 3R regeneration is an issue) maximizes the signal quality. If the network possesses insufficient resources, the setup request is rejected.

Release requests cause all resources associated with a given path to be released and they thus become available for future call setup requests.

6.5.1. Validation and verification

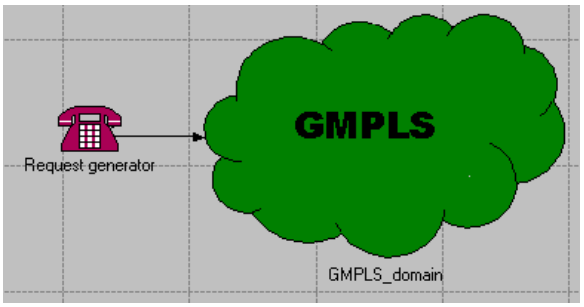


Figure 47: The OPNET test scenario

A number of tests were carried out in order to validate the key routing model. A simple request-generator was implemented to load the model with setup requests.

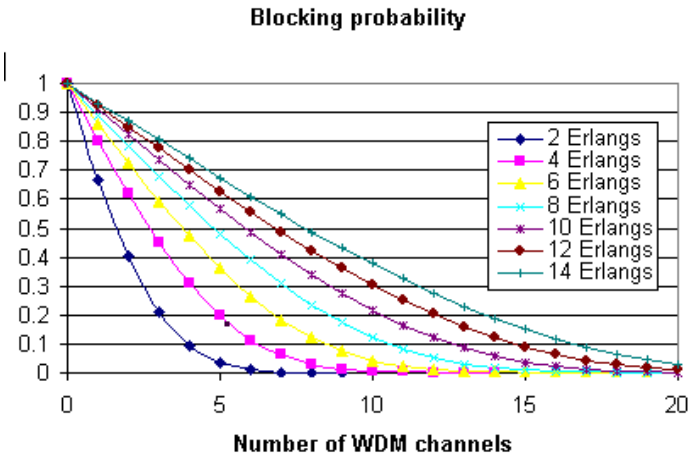


Figure 48: Blocking probability versus number of channels for a simple network consisting of only one link. The parameter is the traffic load on the network.

The results are shown above (Figure 48) and show that the traffic and the number of available WDM channels impacts the setup probability. In order to be able to analytically calculate whether the results are correct, the network topology in this case is the simplest possible: two nodes interconnected by one link.

The results are as expected and since blocked calls are cleared the results can rather easily be checked by using tele traffic theory (the Erlang B formula) since this system is equivalent to a telephone exchange (lost calls cleared) with the number of trunks equal to the number of WDM channels.

6.5.2. Simulation results

The key scheme works! However, what is interesting to see is how the length of the key scales with the network size. Figure 49 depicts the average key length for a certain network size measured in number of nodes and links. Each data point in the graph is the average outcome of 55 simulations and during each simulation 500 call-setup attempts were made. The results have been published in [p08]

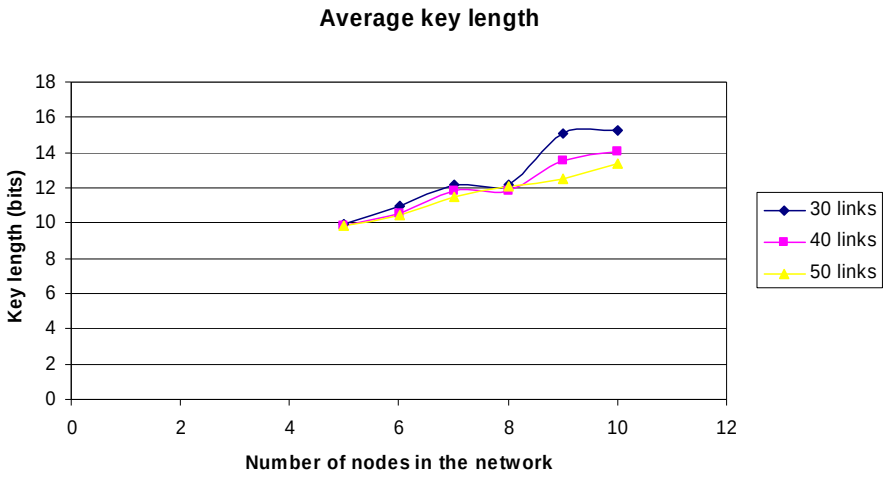


Figure 49: The key length depends on the size of the network

It is shown that a label length of about 2 bytes is sufficient to support network sizes of up to 10 all-optical network nodes. Larger networks will generally require longer paths, which are infeasible without optical regeneration. Clearly the length increases with networks size, but interestingly enough the length is appropriate for optical networks and does not severely impact the use of network resources.

6.5.3. Assessment

The key recognition scheme has a number of advantages. First, since label swapping is not required in order for this scheme to work, lookup tables in the network nodes are avoided completely. This implies that bandwidth can be saved because there is no need for a label distribution protocol to distribute labels, and furthermore the internal switch architecture is simplified greatly. Second, no routing protocol in any core LSR is needed, but there is still a need for some protocol to distribute topology information to the edge nodes so that they can make routing decisions. There are, however, some issues to be considered with respect to this scheme. First of all it is important to employ a function, which can be decoded easily, while enabling edge LSRs to calculate the required key. Furthermore, the routing key can potentially grow to considerable lengths depending on the routing function used as well as on the network size and topology [Wes2001][p14].

In the table below traditional label processing using header erasure/rewriting is compared to the key scheme and another scheme proposed in

[p15][Fjel2000]. Here, the pros and cons of the various schemes are summarized.

Label processing scheme	Lookup tables required?	Bandwidth efficiency?	Processing complexity?	Packet format independence?	Wavelength conversion?
Previous schemes for header erasure / rewriting	Yes	Good	Low	No	No
Label swapping by XOR	Yes	Good	Low	Yes	Yes
Key recognition	No	Medium	Medium	Yes	NA

6.6. Summary

This chapter has highlighted methods for using simulation in a network context. The size and complexity of modern networks mandates considerations on how to simplify the simulation models. Otherwise, the time required to complete the simulations will be too large.

In this chapter mainly modeling of aggregation devices was treated. It was shown how really simple models could be built and how network wide conclusion could be made.

In future, mixed-technology networks adaptation devices will be required to adapt traffic such that the resources available in each technology can be utilized efficiently. An OPNET simulation model of such aggregation units has been described. A variety of scheduling schemes can be used in the aggregation units and the schemes have been evaluated with respect to delay, delay variation and bundling efficiency.

The results showed the scheduling unit as having a significant impact on the traffic characteristics. The undesirable properties of a simple round robin scheme (the unbounded delay) can be avoided by using a more elaborate scheduler. The *'time stamping'* scheme described and analyzed here is shown to give far better performance with regard to delay. This is especially true for realistic input traffic patterns where packets are unevenly distributed among the queues.

TCP traffic on top of such adaptation was modeled by using OPNET modeler. The results showed that the queuing scheme in the adaptation devices had only marginal impact on the application level performance. It would be nice to conclude something regarding future network architectures. However, As stated in the beginning if this chapter, conclusions on a network level based on simple modeling are very difficult to make. Hence, I shall refrain from doing that and instead hope that the results and ideas presented is yet another piece in the total picture of networks.

Furthermore, a new scheme for packet forwarding without table lookups has been presented. The scheme is, in principle, very simple and relies on a mathematical function to link a routing key, contained within the packets, to a unique identifier associated with each switch to give the output port. Since no lookup tables are needed, and a label distribution protocol can also be avoided, the requirements to the processing power of the switches are reduced. This means that a more efficient use of bandwidth is possible.

A GMPLS network with the key routing scheme was modeled in OPNET. The results showed that 16 bits was enough to contain the routing key needed for optical networks of realistic size.

7. Summary and conclusions

"It's a giant leap for a man – but one small step to mankind?"

N. Armstrong / the author

The end-to-end argument behind the Internet states that new functionality should be implemented in the end-nodes rather within the network. However, for heterogeneous network some functionality must be implemented within the network – and from now on networks will be heterogeneous!

A number of requirements to future networks can be set up:

- High capacity
- Transparency (signal format, protocol independence)
- Traffic engineering capability
- End-to-end QoS support
- Flexibility

All of these requirements can be met by using a combination of technologies along with hierarchical MPLS – or H-MPLS, which is a novel scheme for combining all the networks and make them seamlessly interoperate. This might for instance be an excellent way of harnessing the power of optics. Such architecture might be beneficial for heterogeneous networks where technologies with diverse characteristics must be accommodated.

7.1. Mixed technology networks - revisited

By grouping network nodes using the same technology a heterogeneous network can be turned in to a hierarchical one. In this thesis hierarchical network architectures have been investigated. Such architectures require some adaptation devices that can interconnect domains of different technologies, and their design could be crucial in future network architectures. Among other things optical packets switching could be introduced in such architectures.

Modeling and simulation has been used to investigate these adaptation devices in a network context. The size and complexity of modern networks mandates considerations on how to simplify the simulation models. Otherwise, the time required to complete the simulations will be too large. It was shown how really simple models could be built and how network wide conclusion could be made. This concept is called mixed complexity modeling, because, some part of the network is modeled in full detail while the rest is simplified.

The results showed the scheduling unit as having a significant impact on the traffic characteristics. The undesirable properties of a simple round robin scheme (the unbounded delay) can be avoided by using a more elaborate scheduler. The *'time stamping'* scheme described and analyzed here is shown to give far better performance with regard to delay. This is especially true for realistic input traffic patterns where packets are unevenly distributed among the queues. In addition, a preliminary study of TCP on top of such aggregation scheme showed that the its impact on application running over TCP was negligible.

Furthermore, a new scheme for packet forwarding without table lookups has been presented. The scheme is, in principle, very simple and relies on a mathematical function to link a routing key, contained within the packets, to a unique identifier associated with each switch to give the output port. Since no lookup tables are needed, and a label distribution protocol can also be avoided, the requirements to the processing power of the switches are reduced. This means that a more efficient use of bandwidth is possible. Simulation results showed that 16 bits was enough to contain the routing key needed for optical networks of realistic size.

7.2. Transparency

Protocol independence is related to transparency and is one of the major issues when talking network architecture. It seems to be a *fait accompli* that IP will play a role in this scenario. It could be discussed whether one should go along the Internet path and strive to develop one common layer 3 protocol for all current and future applications of the network. The problem is that if such a protocols imposes limitations on the new applications then it might inhibit new applications a hence Internet evolution as such. A better approach might be to use IP as a common adaptation protocols towards the clients and then internally in the network use something else. In this way the network will still appear as a homogeneous network to the user, but in reality this layer-3 transparency is not present in the network. MPLS is a likely candidate due to its ability to provide high speed, scalability and IP operability.

Figure 50 depicts a likely evolution for the evolution of network transparency. Figure 50 depicts a scenario equivalent to the current situation, in which IP networks constitutes one of several possibilities. As time goes by IP is expected to play a bigger role as illustrated in figure b. Eventually, IP will cover the entire network – at least seen from an application point of view. Within the network traffic might transparently to the user be carried on many different technologies. from a) a mixed architecture, in which many protocols share a common physical infrastructure, through b) the IP-centric network, where the majority of network nodes run IP and IP hides the underlying topology and protocols and c) the envisaged, future architecture in which IP is present at the network edge only, acting as a common network layer interface to the core network responsible for bulk transport and build on e.g., optical technologies and use H-MPLS. In this way maximum flexibility can be achieved.

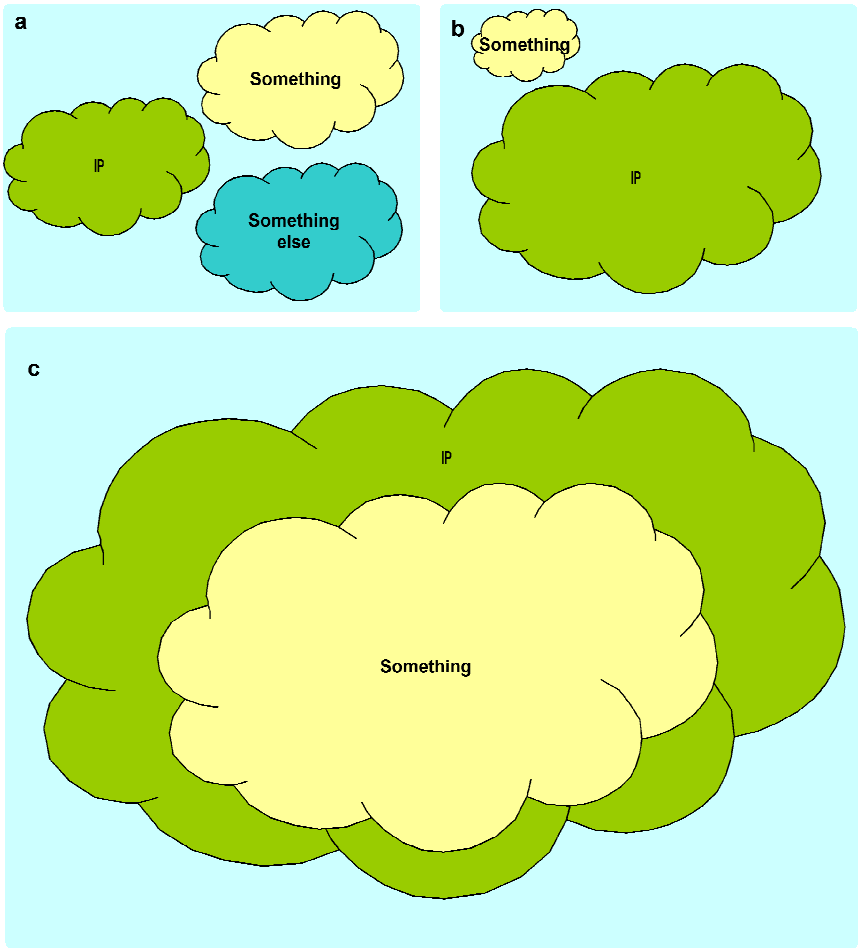


Figure 50: Evolution of the network architecture

8. References

[3GPP922] 3GPP TR 23.922, *"Architecture for an All IP network"*,v.1.0, October 1999.

[3GPP978] 3G TR 21.978 V3.0.0, *"Feasibility Technical Report – CAMEL Control of VoIP Services"*, 3GPP

[Allman2000] M. Allman (editor), *"Ongoing TCP Research Related to Satellites"*, RFC-2760, 2000

[Allman2002] M. Allman, S. Floyd, C. Partridge, *"Increasing TCP's Initial Window"*, RFC-3390, 2002

[And2000] Andersson, L., et. al., *"LDP Specification"*, Internet draft, draft-ietf-mpls-ldp-08.txt, work in progress, June 2000

[AnSIM] http://www.i-u.de/schools/hellbrueck/ansim/

[Ark2000] Arkut, I.C:, Arkut, R.C., Ghani, N., *"Graceful Label Numbering in Optical MPLS Networks"*, proceedings of Opticomm 2000 pp. 1-8, Dallas, USA, October 2000

[Assi2001] C.Assi, Y.Ye, A.Shami, S.Dixit, I.Habib, M.Ali, *"On the Merit of IP/MPLS protection/restoration in IP over WDM networks"*, In proceedings, Global Telecommunications Conference, 2001

[Awd2002] D. Awduche, B, Jabbari, *"Internet traffic engineering using multi-protocol label switching (MPLS)"*, Computer Networks, Vol. 40, issue 1, 2002

[Awe2000] J.Aweya, *"On the design of IP routers. Part 1: Router architectures"*, Journal of Systems Architecture, vol. 46 (2000) pp. 483-511

[Bane2002] G. Banerjee, D. Sidhu, *"Comparative analysis of path computation techniques for MPLS traffic engineering"*, Computer Networks, Vol. 40 issue 1, 2002

[Banks97] J. Banks, R. Gibson, *"10 rules for Determining when Simulation is Not Appropriate"*, IEE Solutions, Issue 29 part 9, 1997

[Ber2002] M. Berger, *"Multipath packet switch using packet bundling"*, proceedings of HPSR, Kobe, Japan, 2002

[Ber2003] M. Berger, H.Christiansen, B. Mortensen, R. Jociles-Ferrier, *"Hierarchical Electro-Optical Packet Network Architecture"*, IST2003, Isfaran, Iran, August, 2003

[Bluehoc] http://www-124.ibm.com/developerworks/opensource/bluehoc/

[Bra1997] Braden, R., et al, *"Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification"*, RFC 2205, September 1997

[Bragg2000] Arnold W. Bragg, *"Which network Design Tool is Right for You?"*, ITProfessional , IEEE Computer Society, September/October 2000.

[CDMASim] http://watt.icu.ac.kr/wins.asp

[Chi1998] Chiaroni, D., Lavigne, B., Jourdan, A., Sotom, M., Hamon, L., et al. *"Physical and logical validation of a network based on all-optical packet switching systems"*, Journal of lightwave technology, vol. 16 # 12, December 1998

[Chi2003] S. Chien, C. Tan, A. Low, A. You, K. Takahashi, *"Buffering controls for IP packets in optical packet switching"*, Proceedings of 9th Asia-Pacific Conference on Communications, 2003

[Clark2002] D. Clark, J. Wroclawski, K. Solins, R.Braden, *"Tussle in Cyber-space: Defining Tomorrow's Internet"*, ACM SIGCOMM Conference, 2002
[CNET] http://www.cs.uwa.edu.au/cnet/
[Col2001] D. Colle, P. Heuven, C. Develder, S. Berghe, I. Lievens, M.Pickavet, P.Demester, *"MPLS recovery mechanisms for IP over WDM networks*", Photonic network Communications, #3, 2001
[Coll2001] D. Colle, A., Groebbens, P. van Heuven, S. De Maeschalck, M. Pickavet, P. Demester, "Porting MPLS recovery techniques to the MPλS paradigm", Optical Networks Magazine, Volume 2 #4, Juliý/August 2001
[CP99] ILOG CPLEX 6.5 – reference manual, ILOG, March 1999
[Cra1998] Crawley, E. et al., *"A Framework for QoS-based Routing in the Internet"*, Request for comment (RFC), 2386, August 1998
[D1] L. Dittmann, H. Christiansen, P. Vogel,, A. Kapovits, *"Project objectives and benchmark measures for next generation core and metro networks"*, NGNI NGPN activity, deliverable D1, 2002. Available online at http://www.ngni-core.net
[D2], Á. Kapovits, P. Stollenmayer, *"Top-down assessment of core and metro networks"*, NGNI NGPN activity, deliverable D2, 2002. Available online at http://www.ngni-core.net
[D3] Paul Vogel et al. ,*"Technology for next generation core and metro networks"*, NGNI NGPN activity, deliverable D3, 2002. Available online at http://www.ngni-core.net
[D4] H. Christiansen, *"Topologies and architectures for next generation core and metro networks"*, NGNI NGPN deliverable D4, October 2002. Available online at http://www.ngni-core.net
[Dan1997] Danielsen, S.L., Mikkelsen, B., Joergensen, C., Durhuus, T. and Stubkjaer, K.E., *"WDM Packet Switch Architectures and Analysis of the Influence of Tuneable Wavelength Converters on the Performance"*, Journal of Lightwave Technology, Vol. 15, No. 2, pp. 219-226, 1997.
[DAVID] See the project web-page at: http://david.com.dtu.dk
[Detti2002] A. Detti, M. Listanti, *"Impact of Segments Aggregation on TCP Reno in Optical Burst Switched networks"*, Proceedings of IEEE Infocom, 2002
[Ditt2001] L. Dittmann, D. Chiaroni, *"DAVID – an approach towards MPLS based optical packet switching with QoS support"*, paper PTHD1, in proceedings of Photonocs in Switching, Monterey, USA, June, 2001
[Ditt2003]L. Dittmann, C. Develder, D. Chiaroni, F. Neri, F. Callegati, Member, IEEE, W. Koerber, A. Stavdas, M. Renaud, A. Rafel, J. Solé-Pareta, W. Cerroni, N. Leligou, Lars Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, and G. Eilenberger, *"The European IST project DAVID: A Viable Approach Toward Optical Packet Switching"*, IEEE Journal on selected Areas in Communication, vol. 11 #7, September 2003
[Elli2003] G. Ellinas, E. Bouillet, R. Ramamurthy, J. Labourdette, S. Chauhuri, K. Bala, *"Routing and restoration architectures in mesh optical networks"*, Optical networks magazine, vol. 4 #1, 2003
[Err1996] Erramilli, A. et al, *"Experimental Queueing analysis with long-range dependent packet traffic"*, IEEE/ACM transactions on Networking, vol. 4 #2, Apirl 1996

[Err1996] Erramilli, A. et al, *"Experimental Queueing analysis with long-range dependent packet traffic"*, IEEE/ACM transactions on Networking, vol. 4 #2, April 1996
[Exten] http://www.eecs.harvard.edu/networking/simulator.html
[Facc2001] S. Faccin and S. Sreementhula, *"Service architecture for next generation networks"*, In proceedings, IEEE Intelligent Network Workshop, 2001
[Feldman] Ph. Feldman, *"Discrete-Event Simulation for Performance Evaluation Systems With Algorithms and Examples in C and C++"* , John Wiley & Sons, 2000.
[Fen2000] C.Fenger, E.Limal, U. Gliese, *"Statistical Study of the influence of Topology on Wavelength Usage in WDM networks"*, In proceeding of Optical Fiber Communication Conference (OFC) 2000
[Fjel2000] T. Fjelde, A. Kloch, D. Wolfson, C. Janz, A. Coquelin, I. Guillemot, F. Gaborit, F. Poingt, F. Dagens, M. Renaud, *"Novel scheme for efficient label-swapping using simple XOR gate"*, Proceedings of ECOC 2000, Vol. 4, paper 10.4.2, pp. 63-64, Munich, Germany, 2000.
[FLAN] http://picolibre.enst-bretagne.fr/projects/flan/
[Flo2001] S. Floyd, V. Paxson, *"Difficulties in Simulating the Internet"*, IEEE/ACM Transactions on Networking, Vol. 9, No. 4, August 2001
[Foi2001] H. Foisel, M. Jaeger, F. Westphal, K. Ovsthus, J Bischoff, *"Evaluation of IP over WDM network architectures"*, Photonic network communication #3, 2001
[Fuji2003] R.M. Fujimoto, K. Perumalla, A. Park, H. Wu, M. Ammar, G.F. Riley, *"Large-scale Network Simulation: How Big? How Fast?"*, Proceedings of the 11th IEEE/ACM International Symposium on modeling, Analysis and Simulation, 2003
[G.709] ITU-T,*"NETWORK NODE INTERFACE FOR THE OPTICAL TRANSPORT NETWORK (OTN)"*, ITU-T recommendation, G.709
[G.803] ITU-T, *"Architecture of transport networks based on the synchronous digital hierarchy (SDH)"*, ITU-T recommendation G.803, March 2000
[G.872] "Architecture of optical transport networks", ITU-T recommendation, G.872
[Gam98] P. Gambini, M. Renaud, C. Guillemot, F. Callegati, I. Andonovic, B. Bostica, D. Chiaroni, G. Corazza, S. L. Danielsen, P. Gravey, P. B. Hansen, M. Henry, C. Janz, A. Kloch, R. Krahenbuhl, C. Raffaelli, M. Schilling, A. Talneau, and Libero Zucchelli, *"Transparent optical packet switching: network architecture and demonstrators in the KEOPS project"*, IEEE Journal on Selected Areas in Communications, Vol. 16 issue 7, 1998
[Gha2000] Ghani, Nashi, "Lambda-labeling: A framework for IP-Over-WDM Using MPLS", Optical networks volume 1 #2, April 2000
[Gha2000] N. Ghani, *"On IP-over-WDM integration"*, IEEE communication magazine, March 2000
[Ghan1999] A. Ghanwani, B. Jamoussi, D. Fedyk, P. Ashwood-Smith, L. Peter, N. Feldman, ,*"MULTIPROTOCOL LABEL SWITCHING - TRAFFIC ENGINEERING STANDARDS IN IP NETWORKS USING MPLS"*, IEEE communication magazine, vol. 37 issue 12, 1999
[Ghani2000] Ghani, Nashi, *"Lambda-labeling: A framework for IP-Over-WDM Using MPLS"*, Optical networks volume 1 #2, April 2000
[GLOMOSIM] http://pcl.cs.ucla.edu/projects/glomosim/

[Gow2003], S. Gowda, R. Shenai, K. Sivalingam, H. Cankaya, *"Performance Evaluation of TCP over Optical burst-switched (OBS) WDM networks"*, proceedings IEEE International Conference on Communications (ICC), 2003
[H.225] ITU-T, *"Call signalling protocols and media stream packetization for packet-based multimedia communication systems"*, ITU-T recommendation, November 2000
[H.323] ITU-T, *"Packet-based multimedia communications systems"*, ITU-T recommendation, November 2000
[Han1998] P. Hansen, S. Danielsen, K. Stubkjaer, *"Optical packet switching without packet alignment"*, Proceedings of ECOC 1998
[Hol2003] J. Holden, C. Chan, *"An investigation of the influence of PRNG properties on Discrete Event Simulation Results Using OPNET modeler"*, OPNETWork 2003
[Hun2000] Hunter, D.K., Andonovic, I., *"Approaches to Optical Internet packet switching"*, IEEE communications Magazine, September 2000
[I.361] ITU-T, "B-ISDN ATM layer specification", ITU-T recommendation, I.361, February, 1999
[IMA97] ATM Forum, *"Inverse Multiplexing for ATM (IMA) Specification"*, Version 1.0, AF-PHY-0086.000, July, 1997
[INSANE] http://www.employees.org/~bmah/Software/Insane/
[Jac1992] V. Jacobson, R. Braden, D. Borman, *"TCP Extensions for High Performance"*, RFC-1323, 1992
[Jain91] R. Jain, *"The Art of Computer Systems Performance Analysis"*, John Wiley & Sons, Inc., 1991, ISBN: 0-471-50336-3
[Jam1999] Jamoussi, B., et. al., *"Constraint-Based LSP Setup using LDP"*, Internet draft, draft-ietf-mpls-cr-ldp-03.txt, work in progress, September 1999
[Krus2001] H. Kruse, M. Allman, J. Griner and D. Tran, *"Experimentation and modelling of HTTP over satellite channels"*, International Journal of Satellite Communications, 2001
[Lel1984] W.E.Leland, M.S.Taqqu, W.Willinger and D.V. Wilson, "On the self-similar Nature of Ethernet traffic (Extended version)", IEEE/ACM transactions on networking, vol. 2 # 1 February 1984
[Lub89] B. Lubachevshy, *"Efficient distributed svent-driven simulation of multiple-loop networks"*, Communication of the ACM, Vol. 32 # 1, 1989
[Mal1998] G.Malkin, *"RIP version 2"*, RFC 2453, November 1998
[Man2001] Mannie, E., (editor.), *"Generalized Multi-Protocol Label Switching (GMPLS) Architecture"*, Internet draft, draft-ietf-ccamp-gmpls-architecture-00.txt, work in progress, June 2001
[Mathis1996] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, *"TCP Selective Acknowledgment Options"*, RFC-2018, 1996
[MET5] O. Marmur, T. Muzicant, H. Christiansen, *"Network Architecture and System Requirements"*, METEOR deliverable D05, 2001
[Met96] R. Metcalfe, *"Packet Communication"*, Peer-to-peer Communications, 1996
[METEOR] see http://www.ist-meteor.org
[Mie99]P. Mieghem, *"Topology information condensation in hierarchical networks"*, Computer Networks, Vol. 31, 1999, ISSN 13891286, pp. 2115 – 2137
[Moy1998] Moy, J., *"OSPF version 2"*, Request for comment (RFC) 2328, april 1998

[MPLSSim] http://flower.ce.cnu.ac.kr/~fog1/mns/
[NCTUns] http://nsl10.csie.nctu.edu.tw/
[NetSIM] http://eewww.eng.ohio-state.edu/drcl/grants/middleware97/netsimQ.html
[NGNI] see http://www.ngni.org/
[Nik2002] E. Nikolouzou et al., *"Network services definition and deployment in a differentiated services architecture"*, in proceedings, IEEE International conference on communication (ICC) 2002
[NIST] http://w3.antd.nist.gov/Hsntg/prd_atm-sim.html
[NS2] http://www.isi.edu/nsnam/ns/
[OPNET] http://www.opnet.com
[Paw2002] K. Pawlikowski, H. Jeong, J. Lee, *"On Credibility of simulation studies of Telecommunication Networks"*, IEEE Communication Magazine, January 2002
[Pio2000] Pióro, M., Stidsen, T., Glenstrup, A., Fenger, C., Christiansen, H., *"Design problems in robust optical networks"*, Session TST-03, 'Networks 2000' , Toronto, September 2000
[Pio97] M. Pióro, *"Solving multicommodity integral flow problems by simulated allocation"*, Telecommunicatio Systems, vol. 7, 1997
[PNNI2002] ATMForum,*"Private Network-Network Interface Specification v.1.1"*, af-pnni-0055.001, April 2002
[QUALNET] http://www.scalable-networks.com/
[REAL] http://www.cs.cornell.edu/skeshav/real/overview.html
[Rek2000] Y. Rekhter, *"Carrying Label Information in BGP-4"*, Internet Draft, draft-ietf-mpls-bgp4-mpls-04.txt, work in progress, January 2000
[Rek95] Rekhter, Y., *"CIDR and Classful Routing"*, Request for comment (RFC) 1817, August 1995
[Rex96] J.L. Rexford, A.G. Greenberg, F.G. Bonomi, *"Hardware efficient Fair queuing Architecture for High-speed networks"*, In proceedings of IEEE INFOCOM, 1996
[rfc1631] K. Egevang, P. Francis, *"The IP Network Address Translator (NAT)"*, Request for comments (RFC) 1631, May 1994
[rfc2002] C. Perkins (Editor), *"IP mobility support"*, Request for comments (RFC) 2002, October 1996
[rfc2131] R. Droms, *"Dynamic Host Configuration Protocol"*, Request for comments (RFC) 2131, March 1997
[rfc2215] S. Shenker, J. Wroclawski, *"General Characterization Parameters for Integrated Service Network Elements"*, Request for comments (RFC) 2215, September 1997.
[RFC2460] S. Deering, R. Hinden, *"Internet Protocol, Version 6 (IPv6) - Specification"*, Request for Comments(RFC) 2460, work in progress
[rfc2475] S. Blake," An Architecture for Differentiated Services", Request for comments (RFC) 2475, December 1998
[rfc2543] M. Handley et al., *"SIP: Session Initiation Protocol"*, Request for Comments (RFC) 2543, March 1999
[rfc3032]Rosen, E.C. et al, *"MPLS Label Stack Encoding"*, RFC 3032, Januar 2001
[rfc3168] *"The Addition of Explicit Congestion Notification (ECN) to IP"*, Request for comments 3168, September 2001

[rfc791] “Internet protocol”, Request for comments (RFC) 791. 1981
[rfc793] “Transmission control protocol”, Request for comments (RFC) 793. 1981
[Rip2002] M. Ripeanu, A. Iamnitchi and I. Foster, *“Mapping the Gnutella network”*, IEEE Internet computing, January 2002
[Riv1990] Rivest, R.L., Cormen, T.H., Leiserson, C.E., *"Introduction to Algorithms"*, MIT Press, 1990
[Riv1990] Rivest, R.L., Cormen, T.H., Leiserson, C.E., *"Introduction to Algorithms"*, MIT Press, 1990
[Ros1999] Rosen, E.C: et al, *"MPLS Label Stack Encoding"*, Internet draft draft-ietf-mpls-label-encaps-07.txt, work in progress, September 1999
[Ros99a] Rosen, E.C. et al, *"Multiprotocol Label Switching architecture"*, Internet draft draft-ietf-mpls-arch-06.txt, work in progress, August 1999
[Ros99b] Rosen, E.C: et al, *"MPLS Label Stack Encoding"*, Internet draft draft-ietf-mpls-label-encaps-07.txt, work in progress, September 1999
[Roy99] E. M. Royer, C.-K. Toh, *“A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks”*, IEEE Personal Communications, April 1999, ISSN 10709916, pp. 46 – 55
[Sar2003] R. Sargent, *“Verification and validation of simulation models”*, Proceedings of the 2003 winter simulation conference, December 2003
[Shep97] C. Partridge and T. J. Shepard, *“TCP/IP Performance over Satellite Links”*, IEEE Network, September/October 1997
[SimMAN] http://www.ifak.fhg.de/kommunik/englisch/SimMan.htm
[Smi2004] J. Smith, S. Nettles, *“Active Networking: One View of the Past, Present, and Future”*, IEEE Transactions on Systems, Man and Cybernetics, Vol. 34, issue 1, 2004
[Ste1999] T.E: Stern, K. Bala, *“Multiwavelength optical networks – a layered approach”*, Addison Wesley Longman, 1999
[Stoica2004] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, S. Surana, *“Internet Indirection Infrastructure”*, IEEE/ACM transactions on networking, Vol. 12 #2, April 2004
[Suu1984] J.W., Suurballe, *“A quick method for finding shortest pairs of disjoint paths”*, Networks, vol. 14 # 2, 1984
[Szy2003] B. K. Szymanski, Y. Liu, R. Gupta, *“Parallel Network Simulation under Distributed Genesis”*, Proceedings of the 17th workshop on Parallel and Distributed Simulation, PADS03, 2003
[Tho97] G. Thomas, *“Multi Channel Input-queuing for high throughput switches”*, electronics letters, vol. 33 #3, 1997
[Tin1989] P.A. Tinker, *“Object creation, messaging, and state manipulation in an object oriented Time Warp system”*, proceedings of the SCS multiconference on Distributed simulation
[VENUS] http://www.bell-labs.com/project/venus/
[Vok2003] V. Vokkarane, J. Jue, *”Burst segmentation: an approach for reducing packet loss in optical burst-switched networks”*, Optical Networks Magazine, November/December, 2003
[Wes2001] Wessing, H., Fjelde, T., Christiansen, H., Dittmann, L., *"Novel scheme for efficient and cost-effective forwarding of packets in optical networks without header modification"*, Proceedings of OFC 2001, paper ThG4_1, Anaheim, USA, 2001

[Will1997] W. Willinger, M. Taqqu, R. Sherman, D. Wilson, *"Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level"*, IEEE/ACM Transactions on Networking, Vol. 5 issue 1, 1997.
[WIPSIM] http://www.wipsim.net/
[Wol1999] Wolfson, D., Hansen, P.B., Kloch, A., Fjelde, T., Janz, C., Coquelin, A., Guillemot, I., Gaboit, F., Poingt, F., Renaud, M., *"All-optical regeneration at 40 Gbit/s in an SOA-based Mach-Zehnder interferometer"*, in Technical Digest of OFC'99, post deadline paper PD36, San Diego, USA, Feb., 1999
[You2003] O. Younis, S. Fahmy, *"Constraint-based Routing IN the Internet: Basic Principles and Recent Research"*, IEEE Surveys, Vol. 5, 2003
[Zang2001] H. Zang, B. Mukherjee, *"Connection management for survivable wavelength-routed WDM mesh networks."*, Optical Networks Magazine, Volume 2 #4, Juliý/August 2001
[Zeg1996] E.W. Zegura, K.L.Calvert, S.Bhattacherjee *"How to model an Internetwork"*, In proceedings IEEE infocom 1996
[Zui2002] J. Zuidweg, *"Next generation intelligent networks"*, Artech house 2002

9. Frequently used abbreviations and acronyms

Acronym	Meaning
3GPP	3rd Generation Partnership Project
ADM	Add Drop Multiplexer
ATM	Asynchronous Transfer Mode
BER	Bit Error Rate
BGP	Border Gateway Protocol
CDMA	Code Division Multiple Access
CSPF	Constrained Shortest Path First
CWDM	Coarse WDM
DAB	Digital Audio Broadcast
DAVID	Data And Voice Integration over DWDM
DVB	Digital Video Broadcast
DWDM	Dense WDM
EDGE	Enhanced Data Rates for GSM evolution
EU	European Union
FDL	Fiber Delay Line
FEC	Forward Equivalence Class
FIFO	First In First Out
FM	Frequency Modulation
FTP	File Transfer Protocol
GMPLS	Generalized MPLS

Acronym	Meaning
GPS	Global Positioning System
GSM	Global System for Mobile
GUI	Graphical User Interface
HAP	High Altitude Platform
HDLC	High Level Data Link Control
HMPLS	Hierarchical MPLS
HSCSD	High-Speed Circuit Switched Data
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPDS	Inmarsat Packet Data System
ITU	International Telecommunication Union
ITU-T	ITU-Telecommunication Sector
KEOPS	Keys to Optical Packet Switching
LAN	Local area network
LDP	Label Distribution Protocol
LP	Linear Programming
LPM	Longest Prefix Match
LSP	Label Switched Path
MAN	Metropolitan area network
MAN	Mobile Access Node
MAN	Mobile Access Node
MEMS	Micro-Electro-Mechanical Systems
MES	Mobile Earth Station
METEOR	Metropolitan Terabit Optical Ring
MPLS	Multi protocol Label switching
NAT	Network Address Translation

Acronym	Meaning
NGN	Next Generation Network
NGNI	Next Generation Network Initiative
NGPN	Next Generation Photonic Networks
NHLFE	Next Hop Label Forwarding Entry
OADM	Optical ADM
OBS	Optical Burst Switching
OE	Optical / Electrical
OEO	Optical / Electrical / Optical
OPNET	Optimum Performance Network Engineering Tool
OPR	Optical Packet Router
OPS	Optical Packet Switching
OSI	Open Systems Interconnections
OSPF	Open Shortest Path First
PDF	Probability Density Function
PDH	Plesiochronous Digital Hierarchy
PEP	Performance Enhancing Proxies
PHB	Per Hop Behavior
PNNI	Private Network Node Interface
POS	PPP Over SONET
PPP	Point-to-Point Protocol
QoS	Quality of Service
RSVP	Resource Reservation Protocol
SAN	Satellite Access Node
SBS	Satellite Base Station
SDH	Synchronous Digital Hierarchy
SDR	Software defined radio
SMTP	Small Message Transfer Protocol
SONET	Synchronous Optical NETwork

Acronym	Meaning
SPF	Shortest Path First
SRLG	Shared Risk Link Group
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic engineering
UMTS	Universal Mobile Telecommunication System
WAN	Wide Area network
WCDMA	Wideband CDMA
WDM	Wavelength Division Multiplexing
WLAN	Wireless Local Area Network
WWW	World Wide Web
XOR	EXclusive Or